第1章 绪 论

对于许多工程问题，虽然已经得到基本方程和边界条件及初始条件，但能用解析方法求解的只是少数方程简单、边界规则的问题，而绝大多数只能通过其他途径求解。随着计算机硬件和软件的发展，数值方法越来越受到人们的青睐，其中有限元法以其方法的统一性和理论的普遍性而独领风骚，已经成为处理各类科学和工程问题的有效方法之一。众多的有限元商用软件，如 ANSYS、NASTRAN、ADINA、ABAQUS 等，它们包含众多单元类型，能求解各类问题，得到广泛的应用。

有限元法的基本思想是"化整为零"。其实这个思想很早就在各个领域被人们采用，老子就说过"道生一，一生二，二生三，三生万物"，他把世间万物全部分解成最基本的"道"。古代数学家将圆用多边形近似，并以此估算圆周率 π 值，达到了 40 位数字的精度。有限元法把一个复杂的结构分解成相对简单的"单元"，各单元之间通过结点相互连接。单元内的物理量由单元结点上的物理量按一定的假设为插得到，这样就把一个复杂结构从无限多个自由度简化为有限个单元组成的结构，只要分析每个单元的力学特性，然后按照有限元法的规则把这些单元"拼装"成整体，就能够得到整体结构的力学特性。由此可见，有限元"化整为零"的思想十分简单明了。但更重要的是，有限元可以建立在严格的数学基础上[1]，成为求解微分方程的标准方法之一，从而它不仅能够解决结构分析问题，也能解决工程中如电磁场、流体力学、热传导、渗流等领域的诸多问题[2]，因为它们在数学上都能用微分方程来描述。

有限元法是一种数值方法，应用性很强，因此在学习的过程中，必要的程序设计、编写和调试能加深理解有限元理论，增加从实际工程出发建立有限元模型的感性认识，极大地提高学习的兴趣和效率。传统的有限元书籍，凡给出具体程序代码的[2-4]，基本上都是以 FORTRAN 为程序设计语言。由于一个完整的有限元计算程序，要包括模型的输入、稀疏矩阵的存储，求解，分析结果的后处理等，导致程序十分庞大难懂，会给初学者的学习带来很大困难，因此有的就只用一个求解单元刚度矩阵的子程序来说明程序的编写，而这样一段程序无法独立编译连接成一个可执行文件，读者就无法验证其算例以获得感性认识，示例效果大大降低。

本书的特色就是采用目前最流行的科学分析计算软件 MATLAB 作为编程环境。MATLAB 的优势在于采用矩阵作为它的基本数据类型，并提供大量实用的矩阵运算函数，可以把有限元理论中用矩阵表示的公式直接写成形式上一样的程序代码，而且还有专门的极易使用的大型稀疏矩阵存储和求解的功能，从而使得一个完整的有限元程序十分简洁明了，方便读者理解和验证，大大提高了示例程序的效果。目前虽也已经有采用 MATLAB 作为程序设计语言的有限元专著[5]出版，并被翻译成中文[6]。但那些书都以介绍 MATLAB 编写的有限元程序为主，几乎不涉及有限元理论。另一本结合 MATLAB 编写的有限元专著[7]的情况有所不同，它更倾向于把有限元作为一种求解偏微分方程的数值方法来介绍，但目前还未见中译本。

1.1 有限元法简介

有关有限元的起源，很多关于有限元的专著[2,3,8,9,10,11]都会提到 Hrenikoff[12]、Courant[13]、McHenry[14]、Newmark[15]、Turner[16]和 Clough[17]等人的工作。事实上，同样也应该提到 Ritz[18]，他在 1909 年曾提出一个求解连续体力学问题的非常有用的近似方法，后人称为 Ritz 法。该方法把待求函数用一组已知的试函数加权和来表示，通过最小势能原理可求解每个试函数的权系数。Ritz 法最基本的缺点就是每一个试函数必须满足指定的边界条件。Courant[13]对 Ritz 法作了重要的推广，他把整个求解区域分成很多三角形的子区域，然后在子区域内假定待求函数为线性函数，把三角形顶点处的函数值作为未知数，并且把边界条件放宽，只要求在边界的有限点上满足，这样，应用 Ritz 法的一个很大的困难被克服了。更为重要的是，原先的 Ritz 法解的精度很大程度上取决于试函数的选取，这就要求应用人员具有丰富的工程经验。

后来人们发现，Courant[13]应用的 Ritz 法其实与若干年后由 Clough[17]提出的有限元法是一致的。当然，在 1960 年有限元获得迅速发展的原因在于该法中大量的数值运算能够由当时正发展的电子计算机来实现，然而在 1943 年，还没有这个工具为 Courant[13]所用。随着计算机硬件和软件的发展，有限元迅速发展，并渐趋成熟，目前它已经被推广至三维问题、非线性问题、时变问题，甚至已经超越了结构分析领域，如流体流动、热传导和电磁场分析等。

应用有限元法进行结构分析时，应把所分析结构物离散成有限个单元，并在每个单元上指定有限个结点，单元通过这些结点连接构成整个有限元模型，用来模拟所分析的实际结构。同时选定所求物理量的结点值，例如将结点位移作为基本未知量，然后对每个单元假设一个简单的插值函数(称为形函数)，近似地表示未知量在单元内的分布规律，再利用变分原理或其他方法，建立单元结点力和结点位移之间的关系，得到一组以结点位移为未知量的代数方程组，从而求解结点的位移分量。一经解出，就可以利用插值函数确定单元内任意一点的位移值。显然，如果单元满足问题的收敛性要求，那么随着单元尺寸的缩小，求解区域内单元数量的增加，解的近似程度将不断改进，最终收敛于精确解。

当然，有限单元法中并不一定要求取结点位移作为基本未知量，也可以取结点内力为未知量，因而，随着所取未知量的不同，有所谓位移法、力法、杂交法和混合法之分。本书采用最为普遍的位移法，介绍结构分析中的有限元法基本理论和方法。

有限元法概念浅显，容易掌握，可以在不同的水平上建立起对该法的理解，既可以通过非常直观的物理解释，也可以建立基于严格的数学分析的理论[11,10,19,20]。它不仅对结构物的复杂几何形状有很强的适应性，也能应用于结构物的各种物理问题，如静力问题、动力问题、非线性问题、热应力问题等。还能处理非均质材料、各向异性材料，以及复杂边界条件等难题。因此，有限元法已经被公认为是工程分析的有效工具，受到普遍重视。

有限元法还有一个特点是它的理论采用矩阵形式表达。这并不利于一般的计算机语言编制计算机程序，因为传统的计算机语言处理的对象是标量，使用矩阵形式的有限元理论时，必须把矩阵形式的公式转换成标量表示的公式。而如果采用 MATLAB，这个特点就变成了有限元法的优点。这也是本书采用 MATLAB 作为编程环境的一个重要原因。

1.2 MATLAB 简介

MATLAB 是当今国际科学界最具影响力和活力的软件。它起源于矩阵运

算，并已经发展成一种高度集成的计算机语言。它提供了强大的科学计算、灵活的程序设计流程、高质量的图形可视化与界面设计、便捷的与其他程序和语言接口的功能。MATLAB 在各国高校与研究单位发挥着重要的作用。

MATLAB 语言的首创者 Cleve Moler 教授，在数值分析特别是在数值线性代数的领域中很有影响，他参与编写了数值分析领域两个重要的 FORTRAN 程序包 EISPACK 和 LINPACK。他曾在密歇根大学、斯坦福大学和新墨西哥大学任数学与计算机科学教授。1980 年前后，当时的新墨西哥大学计算机系主任 Moler 教授在讲授线性代数课程时，发现了用其他高级语言编程极为不便，便构思并开发了 MATLAB(MATrix LABoratory,即矩阵实验室),这一软件利用了当时数值线性代数领域最高水平的 EISPACK 和 LINPACK 两大软件包中可靠的子程序，用 FORTRAN 语言编写了集命令翻译和科学计算于一身的一套交互式软件系统。

所谓交互式语言,是指人们给出一条命令,立即就可以得出该命令的结果。该语言无需像 C 和 FORTRAN 语言那样,首先要求使用者去编写源程序,然后对其进行编译、连接,最终形成可执行文件。这无疑会给使用者带来极大的方便。早期的 MATLAB 是用 FORTRAN 语言编写的,只能作矩阵运算;绘图也只能用极其原始的方法,即用星号描点的形式画图;内部函数也只提供了几十个。但即使其当时的功能十分简单,当它作为免费软件出现以来,还是吸引了大批的使用者。

后来，Cleve Moler 和 John Little 等人成立了 MathWorks 公司，Cleve Moler 一直任该公司的首席科学家。该公司于 1984 年推出了第一个 MATLAB 的商业版本。当时的 MATLAB 版本已经用 C 语言作了完全的改写，其后又增添了丰富多彩的图形图像处理功能、多媒体功能、符号运算和与其他流行软件的接口功能,使得 MATLAB 的功能越来越强大。

MathWorks 公司于 1992 年推出了具有划时代意义的 MATLAB 4.0 版本，并于 1993 年推出了其微机版，可以配合 Microsoft Windows 一起使用,使之应用范围越来越广。1994 年推出的 4.2 版本扩充了 4.0 版本的功能，尤其在图形界面设计方面更提供了新的方法。

1997 年推出的 MATLAB 5.0 版允许了更多的数据结构，如单元数据、数据结构体、多维矩阵、对象与类等，使其成为一种更方便编程的语言。1999 年初推出的 MATLAB 5.3 版在很多方面又进一步改进了 MATLAB 语言的功能。

2000 年 10 月底推出了其全新的 MATLAB 6.0 正式版(Release 12)，在核心数值算法、界面设计、外部接口、应用桌面等诸多方面有了极大的改进。最近又推出了 MATLAB 7.0。

虽然 MATLAB 语言是计算数学专家倡导并开发的，但其普及和发展离不

开自动控制领域学者的贡献，甚至可以说，MATLAB语言是被自动控制领域学者和工程技术人员捧红的，因为在MATLAB语言的发展过程中，许多有代表性的成就和控制界的要求与贡献是分不开的。迄今为止，大多数工具箱也都是控制方面的。MATLAB具有强大的数学运算能力、方便实用的绘图功能及语言的高度集成性，它在其他科学与工程领域的应用也越来越广，并且有着更广阔的应用前景和无穷无尽的潜能。

“工欲善其事，必先利其器”。如果说有一种十分有效的工具能解决在教学与研究中遇到的问题，那么MATLAB语言正是这样的一种工具。它可以将使用者从繁琐、无谓的底层编程中解放出来，把有限的宝贵时间更多地花在解决问题中，这样无疑会提高工作效率。

目前，MATLAB已经成为国际上最流行的科学与工程计算的软件工具。现在的MATLAB不仅是一个“矩阵实验室”，已成为一种具有广泛应用前景的全新的计算机高级编程语言，被称为“第四代”计算机语言，它在国内外高校和研究部门正扮演着重要的角色。MATLAB语言的功能也越来越强大，能不断适应新的要求提出新的解决方法。可以预见，在科学运算、自动控制与科学绘图领域MATLAB语言将长期保持其独一无二的地位。

1.3 本书的目的和内容

本书的目的是把结构分析有限元法的基本概念、基本理论和基本的程序设计方法介绍给土木工程专业的高年级本科生和研究生。然而，由于各种概念是以非常简单的形式给出的，所以对其他背景的学生和实际工程人员也是有帮助的，同时也适合于那些想把有限元应用于解决实际工程问题的各类人员。

本书内容涉及杆系单元、平面单元、三维空间单元、等参数单元、板壳单元，包括静力问题、动力问题、稳定问题，并给出了丰富的以土木工程，尤其是桥梁与隧道工程为背景的算例，可以帮助读者增加如何从工程实际抽象到有限元模型的感性认识。

参考文献

[1] Brenner S. C., Scott L. R.. The Mathematical Theory of Finite Element Methods, Springer-Verlag, 1993.

[2] Zienkiewicz O. C. The Finite Element Method, Third Edition. McGraw-Hill Inc. ,1997.

[3] 王勖成. 有限单元法. 北京:清华大学出版社,2003.

[4] 凌道盛,徐兴. 非线性有限元及程序. 杭州:浙江大学出版社,2004

[5] Kattan P. I. . MATLAB Guide to Finite Element. Springer-Verlag,2003.

[6] Kattan P. I. . 韩来彬,译. MATLAB 有限元分析与应用. 北京:清华大学出版社,2003.

[7] Kwon Y. W. , Bang H. . The Finite Element Method Using MATLAB, CRC Press, Inc. Boca Raton, Florida,1997.

[8] Bathe K. J. , Wilson E. L. . Numerical Methods in Finite Element Analysis. Prentice-Hall. Inc. ,1976.

[9] Cook R. D. . Concepts and Applications of Finite Element Analysis. John Wiley & Sons Inc. ,1982.

[10] Logan D. L. . A First Course in the Finite Element Method. Second Edition. PWS Publishing Company. ITP,1993.

[11] 龙驭球,龙志飞,岑松. 新型有限元论. 北京:清华大学出版社,2004.

[12] Hrenikoff A. . Solution of problems in elasticity by the framework method. J. Appl. Mech,1941:169-175.

[13] Courant R. . Variational methods for the solution of problems of equilibrium and vibration. Bull. Am. Math. Soc. ,1943:1-23.

[14] McHenry D. . A lattice analogy for the solution of plane stress problems. J. Inst. Civ. Eng. ,1943: 59-82.

[15] Newmark N. M. . Numerical methods of analysis in bars plates and elastic bodies. Numerical Methods in Analysis in Engineering (ed. L. E. Grinter). Macmiilan,1997.

[16] Turner M. J. , Clough R. W. , Matrin H. C. and Topp L. J. . Stiffness and deflection analysis of complex structures, J. Aero. Sci. ,1956: 8-23.

[17] Clough R. W. . The finite element in plane stress analysis. Proc. 2nd ASCE Conf. on Electronic Computation. Pittsburgh, Pa. Sept. ,1960.

[18] Ritz W. . Über eine neue Methode zur Lösung gewissen Variations-Probleme der mathematischen Physik. J. Reine Angew Math. ,1909:1-61.

[19] 丁皓江,何福保,谢贻权,徐兴. 弹性和塑性力学中的有限单元法. 北京:机械工业出版社,1989.

[20] Reddy J. N. . An Introduction to the Finite Element Method. Second Edition. McGraw-Hill,1993.

[21] Bathe K. J. . Finite Element Procedures in Engineering Analysis. Prentice-Hall Inc. , 1982.

第2章 有限元法的预备知识

本章主要介绍有限元法的一些预备知识，包括矩阵和线性代数的基本概念，MATLAB编程基础，弹性力学中的变分原理和能量原理基础以及有限元法的基本过程。如果对这些内容十分熟悉，可以跳过本章，直接阅读第3章。

2.1 矩阵、线性代数和 MATLAB

在数学上，有限元法通过变分法或加权残数法把微分方程化为近似的代数方程组进行求解。如果采用矩阵形式进行中间过程的推导，就能使有限元公式简洁紧凑。为此，必须掌握矩阵和线性代数的基本知识，详细了解矩阵运算的方法，这非常有助于阅读和理解有限元法的基本理论并组织有效的算法，这一节的目标是扼要地介绍以后要用到的矩阵和线性代数的基本知识，事实上，这些都是非常基础的。如果希望了解更详细的内容，可以参考有关矩阵和线性代数的专著[1-3]

在讲述矩阵、线性代数基本知识的同时，还将介绍使用 MATLAB实现矩阵定义、运算等基本操作，这有助于增加感性认识，加深理解。当然，这里给出的 MATLAB 语言的用法都是最基本的，详细的说明可以阅读 MATLAB 软件附带的"帮助"文档，如图 2-1 所示。

另外，在因特网上有许多关于介绍 MATLAB 的专著和网站，本章后面的参考文献中就列出了一些比较著名的网站，读者也可以用网络搜索工具，比如 Google 等在因特网上查找更新更多的有

关 MATLAB 的介绍和说明资料[4-20]。

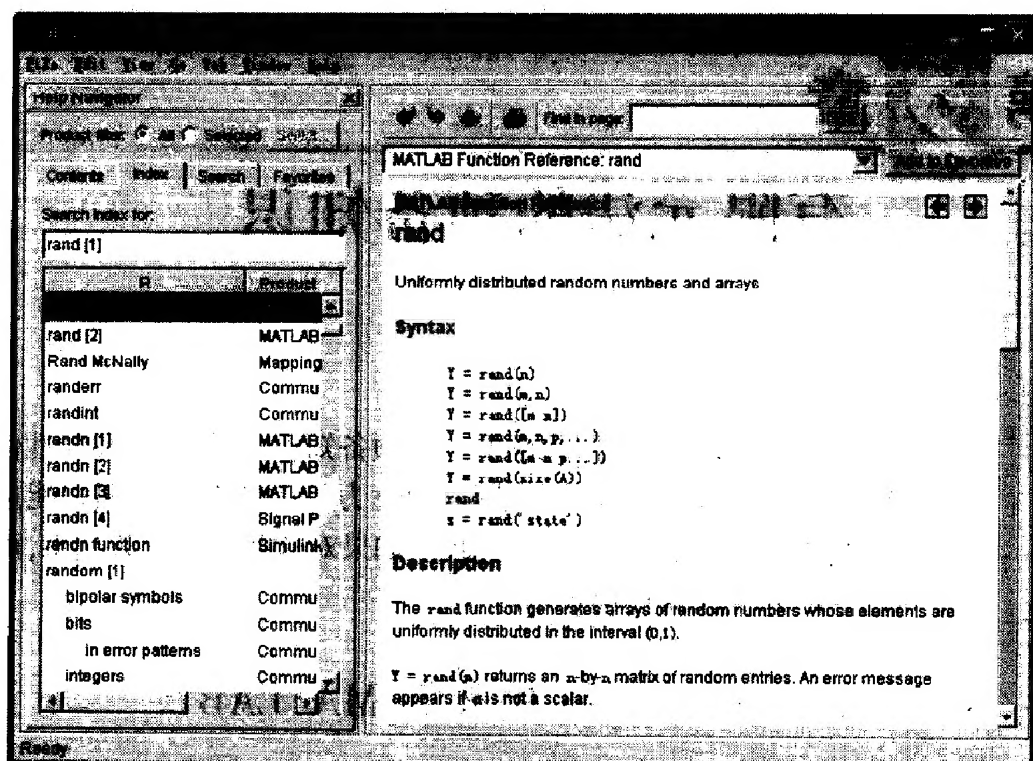


图 2-1 MATLAB 的"帮助"窗口

2.1.1 矩阵简介

在实际计算中矩阵的表达效率可以从求解线性方程组的过程中看出，例如

$$
\left.\begin{aligned}
1.4\delta_1 + 0.0\delta_2 - 1.0\delta_3 - 0.2\delta_4 &= 1.0 \\
0.0\delta_1 + 1.4\delta_2 - 0.4\delta_3 + 0.4\delta_4 &= 0.0 \\
-1.0\delta_1 - 0.4\delta_2 + 1.8\delta_3 + 0.6\delta_4 &= 0.0 \\
-0.2\delta_1 + 0.4\delta_2 - 0.6\delta_3 + 2.4\delta_4 &= 0.0
\end{aligned}\right\} \tag{2-1}
$$

式中，未知数为 $\delta_1,\delta_2,\delta_3$ 和 δ_4，使用矩阵符号，这组方程可写成

$$
\begin{bmatrix} 1.4 & 0.0 & -1.0 & -0.2 \\ 0.0 & 1.4 & -0.4 & 0.4 \\ -1.0 & -0.4 & 1.8 & 0.6 \\ -0.2 & 0.4 & -0.6 & 2.4 \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{Bmatrix} = \begin{Bmatrix} 1.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{Bmatrix} \tag{2-2}
$$

在这里，把未知数的系数按一定的顺序排成一个阵列，左边的未知数（δ_1

$\delta_2,\delta_3,\delta_4$）和右边的已知数也各自被排成一列。虽然写法不同，但式（2-1）和式（2-2）表示了同样的内容。可以进一步把式（2-2）写成

$$A\delta = b \tag{2-3}$$

式中，A 是线性方程组的系数矩阵，δ 是未知数矩阵，b 是已知数矩阵，即

$$A = \begin{bmatrix} 1.4 & 0.0 & -1.0 & -0.2 \\ 0.0 & 1.4 & -0.4 & 0.4 \\ -1.0 & -0.4 & 1.8 & 0.6 \\ -0.2 & 0.4 & -0.6 & 2.4 \end{bmatrix}, \quad \delta = \begin{Bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{Bmatrix}, \quad b = \begin{Bmatrix} 1.0 \\ 0.0 \\ 0.0 \\ 0.0 \end{Bmatrix} \tag{2-4}$$

显然，式（2-3）比式（2-1）或式（2-2）简洁得多。而且，可以任意指定矩阵 A 和未知数 δ 及已知数 b 的阶数，用来表示任意阶数的线性方程组。现在矩阵的正式定义看上起非常明显了，即矩阵是按一定顺序排列的数据阵列。

在 MATLAB 中，定义矩阵变量十分方便和直观，例如式（2-4）中的矩阵 A，可以这样定义：

```
>>A=[ 1.4  0.0 -1.0 -0.2
      0.0  1.4 -0.4  0.4
     -1.0 -0.4  1.8  0.6
     -0.2  0.4 -0.6  2.4 ]
```

这里的符号“ >> ”是 MATLAB 的命令提示符，表示等待输入命令状态。输入上述命令后，在 MATLAB 的命令窗口立刻显示：

```
A =

    1.4000         0   -1.0000   -0.2000
         0    1.4000   -0.4000    0.4000
   -1.0000   -0.4000    1.8000    0.6000
[illegible]
```

表示刚才的输入正确，变量 A 已经被定义成式（2-4）中的矩阵 A。对矩阵 A 中元素的引用也十分方便，比如：

```
[illegible]
```

就得到了矩阵 A 的第 2 行第 3 列元素。矩阵 b 的定义可以这样：

```
>>b=[1.0
     0.0
     0.0
     0.0]
```

也可以只用一行,即

```
>> b=[1,0;0.3;0,0;[illegible]
```

因为在矩阵定义式中,分号";"等价于换行符,输入上面两个命令中的任意一个,MATLAB 的命令窗口都显示:

```
b=
[illegible]
```

因为这两个命令是等价的。

一般的矩阵有 $m\times n$ 个数据,被排成 m 行和 n 列。例如,

$$
A_{m\times n}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}=[a_{ij}]_{m\times n} \tag{2-5}
$$

则说这个矩阵是 $m\times n$ 阶矩阵,在不引起歧义的情况下,可简写为 $\boldsymbol{A}$。当矩阵只有一行($m=1$)或一列($n=1$)时,称它为行向量或列向量。本书统一用大写的黑体字母表示矩阵或行向量,用小写的黑体字母表示列向量。另外,用 a_{ij} 表示矩阵 $\boldsymbol{A}$ 的第 i 行第 j 列的元素。

2.1.2 特殊矩阵

1. 零矩阵

元素全部为零的 $m\times n$ 阶矩阵,称为零矩阵,记为 $\boldsymbol{0}_{m\times n}$ 或者简写为 $\boldsymbol{0}$。

MATLAB 中定义零矩阵十分方便,比如定义一个 4×3 的零矩阵的命令是:

```
>> A=zeros(4,3)
A=
  0  0  0
  0  0  0
  0  0  0
```

命令 zeros 可以接受任意个参数,但一般使用的都是 2 维的矩阵,前一个参数表示行数,后一个是列数。

2. 对称矩阵

我们把 $m \times n$ 阶矩阵 $\boldsymbol{A}$ 的转置，记为 $\boldsymbol{A}^{\mathrm{T}}$，它由 $\boldsymbol{A}$ 中的行和列互换得到。如果 $m=n$，则称 $\boldsymbol{A}$ 为 n 阶方阵。如果 n 阶方阵 $\boldsymbol{A}$ 的元素 $a_{ij}=a_{ji}$，则称 $\boldsymbol{A}$ 为对称矩阵。因此对称矩阵必须是方阵。如果 $\boldsymbol{A}$ 是对称矩阵，则有 $\boldsymbol{A}=\boldsymbol{A}^{\mathrm{T}}$。显然，式(2-4)中的矩阵 $\boldsymbol{A}$ 是对称矩阵。例如设

$$\boldsymbol{A}=\begin{bmatrix} -1 & 2 & 3 \\ 4 & -3 & 1 \end{bmatrix}$$

那么

$$\boldsymbol{A}^{\mathrm{T}}=\begin{bmatrix} -1 & 4 \\ 2 & -3 \\ 3 & 1 \end{bmatrix} \tag{2-7}$$

MATLAB 的函数 transpose 是专门对矩阵进行转置操作的，例如：

```
>> A=[-1   2  3
      4  -3  1]
A =
   -1   2   3
    4  -3   1
>> transpose(A)
ans =
   -1   4
    2  -3
    3   1
```

函数 transpose 可以用一个对应的操作符单引号“'”来代替，例如：

```
>> A'
ans =
   -1   4
    2  -3
    3   1
```

这里需要指出的是，函数 transpose 与单引号“'”对于实矩阵的操作是完全一样的，而对于复矩阵则不同。因为复矩阵在数学上的转置定义不仅包括行列互换，还要执行共轭操作，函数 transpose 只执行行列互换的操作，而单引号则是数学意义上的转置。

3. 单位矩阵

另一个特殊矩阵是单位矩阵 $\boldsymbol{I}_n$，它是 n 阶方阵，除了对角线上的元素为 1

外，其余元素都为零。例如，一个3阶的单位阵为

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2-8}$$

在实际计算时，单位矩阵的阶数常常是隐含的。与单位矩阵相似，我们定义一个 n 阶的单位列向量 e_i，式中 i 表示该向量是单位矩阵中的第 i 列。

MATLAB 中用 eye 来定义单位矩阵。也许读者可能奇怪为什么不用 I，这是因为小写的 i 已经被用来定义复数单位，而大写的 I 容易引起混淆，而且 MATLAB 中内部的变量和函数名全部是用小写的。下面的命令定义了一个 3 阶的单位矩阵。

```
>> eye(3)
ans =
   1 0 0
   0 1 0
```

函数 eye 也能接受多个参数，当它们不相等时，输出的是对角线元素为 1，其余元素为零的非方阵。例如：

```
>> eye(5,4)
ans =
   1 0 0 0
   0 1 0 0
   0 0 1 0
   0 0 0 1
   0 0 0 0
>> eye(3,4)
ans =
   1 0 0 0
   0 1 0 0
   0 0 1 0
```

4. 带状稀疏矩阵

在有限元中，将会碰到大量的对称带状稀疏矩阵，因为一般情况下，刚度矩阵和质量矩阵都是带状稀疏矩阵。带状矩阵是指矩阵中位于带宽以外的元素全为零。如果带状矩阵 A 是对称的，可以把这种情况用等式表示为

$$a_{ij} = 0 \text{ for } j > i + HBW \tag{2-9}$$

对于式(2-9)，$2HBW+1$ 是矩阵 A 的带宽。作为一个例子，下述矩阵是

个 6 阶的对称带状矩阵,它的半带宽 *HBW* 是 2:

$$
\boldsymbol{A}=\begin{bmatrix}8&3&2&0&0&0\\3&7&2&1&0&0\\2&2&9&2&2&0\\0&1&2&7&2&1\\0&0&2&2&9&4\\0&0&0&1&4&8\end{bmatrix} \tag{2-10}
$$

如果一个矩阵的半带宽为零，则该矩阵只有位于对角线的元素非零，此时称它为对角矩阵。例如单位矩阵就是对角矩阵。

如果用普通的矩阵变量来存储这种矩阵，会浪费大量的空间，因为稀疏矩阵中的很多零元素不参加运算，也就不用存储。由于计算机的内存有限，因此必须合理地安排存储空间，使得在计算中要用到的数据都能放入内存，否则频繁地读取外存，将使计算速度大大降低。对于这种带状稀疏矩阵，应该组织一种合理的存储方法，使得内存中能容下高阶稀疏矩阵。图 2-2 所示为一对称带状稀疏矩阵，该矩阵在轮廓线以外的元素为零，而且在以后的运算中永远为零，因此不必存储。而在轮廓线以内的元素必须存储，即使某一元素可能暂时为零，但是经过运算后就会变成非零，因此一定要存储。它的一维存储方法如图 2-2 所示。

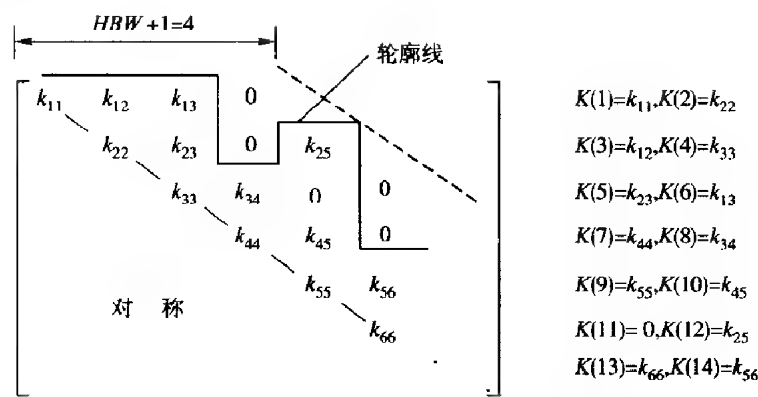


图 2-2 对称带状矩阵 *K* 的一维存储

在有限元程序中，整体刚度矩阵就属于这类矩阵，它们阶数很大，但是大部分元素为零。在 MATLAB 中，有专门处理这类矩阵的工具，从而可以极大地节省处理稀疏矩阵的时间，使使用者可以绕开很多烦人的细节而专心于有限元程序设计的主要方面。这对于初学者尤其重要，因为对于初学者来说，这些问题往往是他们编写有限元程序的主要障碍，而不是掌握有限元的基本理

论。但是目前的 **MATLAB** 中，稀疏矩阵还存在一个缺点，就是不能指定它是对称的，因此会浪费不少内存。

在 **MATLAB** 中,可以用命令 sparse 定义一个稀疏矩阵，比如：

```
>> A = sparse(4000, 4000)
```

屏幕上立刻显示：

```
A =
   All zero sparse: 4000-by-4000 [illegible]
```

以上操作表示 A 是一个 4000×4000 个元素全部为零的稀疏矩阵。稀疏矩阵元素的引用跟普通的矩阵一样。

2.1.3 矩阵的运算

1. 相等

设有两个 $m \times n$ 矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$,如果它们对应的元素都相等,即

$$a_{ij} = b_{ij} \qquad (i = 1,2,\cdots,m; j = 1,2,\cdots,n) \tag{2-11}$$

则称 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 相等,记为 $\boldsymbol{A} = \boldsymbol{B}$。

2. 加法与减法

设有两个 $m \times n$ 矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$,它们相加可以得到一个 $m \times n$ 矩阵 $\boldsymbol{C}$,即 $\boldsymbol{C} = \boldsymbol{A} + \boldsymbol{B}$ 而矩阵 $\boldsymbol{C}$ 的每一个元素由下式确定：

$$c_{ij} = a_{ij} + b_{ij} \qquad (i = 1,2\ \cdots,m; j = 1,2\ \cdots,n) \tag{2-12}$$

如果另有一个 $m \times n$ 矩阵 $\boldsymbol{D} = [-b_{ij}]_{m \times n}$,则称矩阵 $\boldsymbol{D}$ 为矩阵 $\boldsymbol{B}$ 的负矩阵,记为 $\boldsymbol{D} = -\boldsymbol{B}$。从而可以定义矩阵的减法为

$$\boldsymbol{A} - \boldsymbol{B} = \boldsymbol{A} + (-\boldsymbol{B}) \tag{2-13}$$

例如有矩阵

$$\boldsymbol{A} = \begin{bmatrix} -1 & 2 & 3 \\ 4 & -3 & 1 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} -1 & 1 & -1 \\ -3 & 2 & -5 \end{bmatrix} \tag{2-14}$$

那么它们的和与差分别是

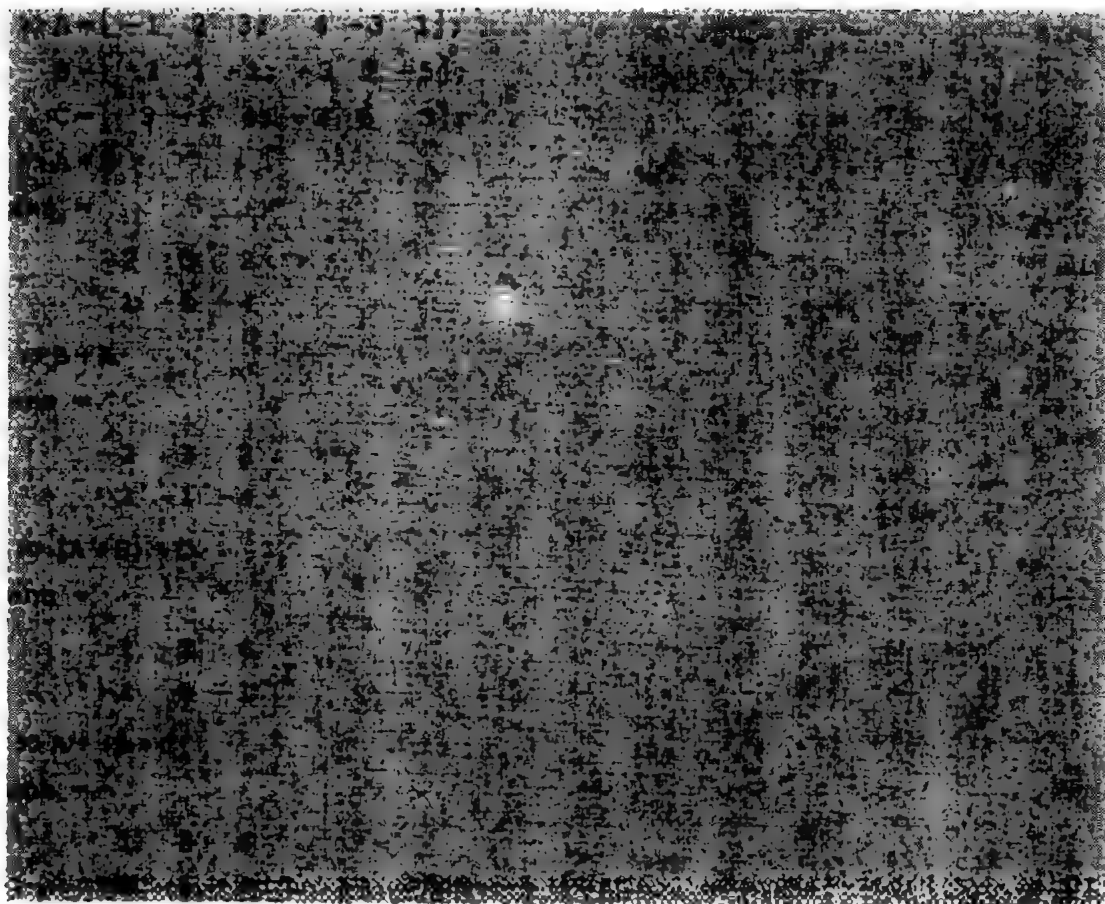
$$\boldsymbol{A} + \boldsymbol{B} = \begin{bmatrix} -1-1 & 2+1 & 3-1 \\ 4-3 & -3+2 & 1-5 \end{bmatrix} = \begin{bmatrix} -2 & 3 & 2 \\ 1 & -1 & -4 \end{bmatrix}$$

$$\boldsymbol{A} - \boldsymbol{B} = \begin{bmatrix} -1+1 & 2-1 & 3+1 \\ 4+3 & -3-2 & 1+5 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 4 \\ 7 & -5 & 6 \end{bmatrix} \tag{2-16}$$

矩阵的加法和减法满足交换律和结合律，即设 $\boldsymbol{A}, \boldsymbol{B}$ 和 $\boldsymbol{C}$ 都是 $m \times n$ 阶矩阵,有下列公式成立

$$\boldsymbol{A} + \boldsymbol{B} = \boldsymbol{B} + \boldsymbol{A} \quad (\boldsymbol{A} + \boldsymbol{B}) + \boldsymbol{C} = \boldsymbol{A} + (\boldsymbol{B} + \boldsymbol{C}) \tag{2-17}$$

MATLAB 中矩阵的运算与单个数值的运算具有相同的形式，比如：



可见 MATLAB 对矩阵的操作十分方便。

3. 数乘矩阵

设 $\boldsymbol{A}$ 是 $m \times n$ 阶矩阵，k 是一个常数，如果另有一个 $m \times n$ 阶 $\boldsymbol{C} = k\boldsymbol{A}$，则 $c_{ij} = ka_{ij}$。例如，如果

$$\boldsymbol{A} = \begin{bmatrix} -1 & 2 & 3 \\ 4 & -3 & 1 \end{bmatrix} \tag{2-18}$$

那么

$$2\boldsymbol{A} = \begin{bmatrix} -2 & 4 & 6 \\ 8 & -6 & 2 \end{bmatrix} \tag{2-19}$$

数乘矩阵运算也满足交换律、结合律和分配律。设 k, l 为常数，$\boldsymbol{A}$ 和 $\boldsymbol{B}$ 为 $m \times n$ 阶矩阵，则有

$$k\boldsymbol{A} = \boldsymbol{A}k, \quad k(l\boldsymbol{A}) = (kl)\boldsymbol{A}, \quad (k+l)\boldsymbol{A} = k\boldsymbol{A} + l\boldsymbol{A} \quad k(\boldsymbol{A}+\boldsymbol{B}) = k\boldsymbol{A} + k\boldsymbol{B} \tag{2-20}$$

下面用 MATLAB 来验证公式(2-20)

```
>> A = [-1 2 3; 4 -3 1];
   B = [-1 1 -1; -2 6 3];
   k = 4;
   l = 2;
>> k* A
ans =
    -4     8    12
    16   -12     4
>> A* k
ans =
    -4     8    12
    16   -12     4
>> k* (l* A)
ans =
    -8    16    24
    32   -24     8
>> (k* l)* A
ans =
    -8    16    24
    32   -24     8
>> (k + l)* A
ans =
    -6    12    18
    24   -18     6
>> k* A + l* A
ans =
    -6    12    18
    24   -18     6
>> k* (A + B)
ans =
    -8    12     8
     8    12    16
>> k* A + k* B
ans =
    -8    12     8
     8    12    16
```

4. 矩阵与矩阵相乘

如果矩阵 $\boldsymbol{A}$ 的列数与矩阵 $\boldsymbol{B}$ 的行数相等，则可以定义矩阵 $\boldsymbol{A}$ 与矩阵 $\boldsymbol{B}$ 的乘法。设 $\boldsymbol{C}=\boldsymbol{AB}$ ，如果 $\boldsymbol{A}$ 的阶数为 $p\times m$，矩阵 $\boldsymbol{B}$ 的阶数为 $m\times q$，那么矩阵 $\boldsymbol{C}$ 的阶数为 $p\times q$。而矩阵 $\boldsymbol{C}$ 的每一个元素定义为

$$c_{ij} = \sum_{r=1}^{m} a_{ir} b_{rj} (i = 1,2,\cdots,p, j = 1,2,\cdots,q) \tag{2-21}$$

式(2-21)的含义是矩阵 $\boldsymbol{C}$ 的第 i 行第 j 列元素等于矩阵 $\boldsymbol{A}$ 的第 i 行和矩阵 $\boldsymbol{B}$ 的第 j 列各对应元素积的和。由此可以看出两个矩阵相乘共有 $p\times m\times q$ 次乘法，计算比较费时。

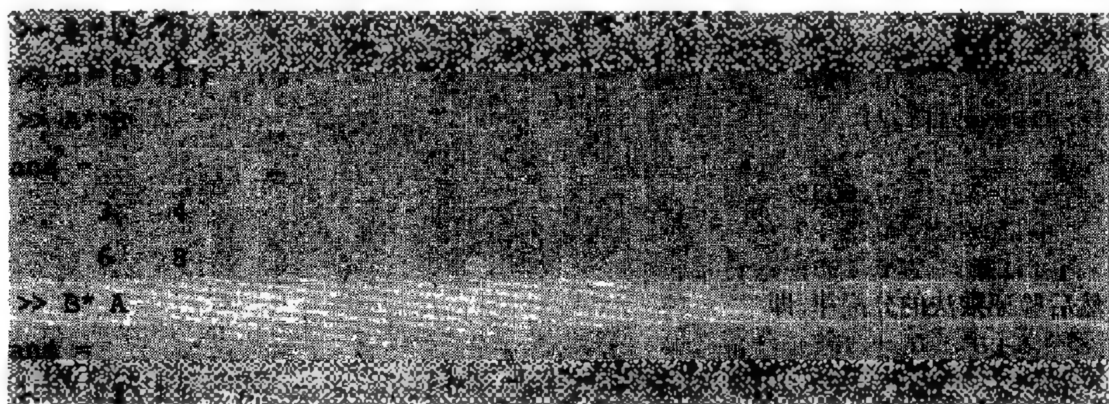
众所周知，常数相乘可以交换，即，$ab=ba$。现在需要研究矩阵相乘是否也满足交换，为此考虑两个矩阵

$$\boldsymbol{A}=\begin{bmatrix}1\\2\end{bmatrix},\quad \boldsymbol{B}=[3\quad 4] \tag{2-22}$$

则有

$$\boldsymbol{AB}=\begin{bmatrix}3 & 4\\6 & 8\end{bmatrix},\quad \boldsymbol{BA}=11 \tag{2-23}$$

用 MATLAB 来验证式(2-23)，即



因此，$\boldsymbol{AB}$ 和 $\boldsymbol{BA}$ 的结果是不同的。也就是说，矩阵相乘不满足交换律。实际上，由于矩阵相乘依赖于矩阵的阶数，当 $\boldsymbol{AB}$ 能进行计算时，$\boldsymbol{BA}$ 可能没有定义。

为了区分矩阵相乘的的顺序，通常称 $\boldsymbol{AB}$ 为矩阵 $\boldsymbol{A}$ 前乘(左乘)矩阵 $\boldsymbol{B}$，也可称为矩阵 $\boldsymbol{B}$ 后乘(右乘)矩阵 $\boldsymbol{A}$。虽然一般情况下 $\boldsymbol{AB}\neq\boldsymbol{BA}$，但是在特殊情况下 $\boldsymbol{AB}=\boldsymbol{BA}$，此时则称 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 为可交换的。

虽然矩阵相乘不满足交换律，但是满足分配律和结合律。设矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 有

相同的阶数 $p \times m$，它们能与阶数为 $m \times q$ 的矩阵 C 相乘，那么分配律可表述为

$$(A+B)C = AC + BC \tag{2-24}$$

上式可很容易地证明如下

$$\sum_{r=1}^{m}(a_{ir}+b_{ir})c_{rj} = \sum_{r=1}^{m} a_{ir}c_{rj} + \sum_{r=1}^{m} b_{ir}c_{rj} \quad (i=1,2,\cdots,p;j=1,2,\cdots,q) \tag{2-25}$$

如果矩阵 A 能与矩阵 B 相乘，矩阵 B 能与矩阵 C 相乘，那么结合律表述如下

$$(AB)C = A(BC) = ABC \tag{2-26}$$

下面用 MATLAB 验证矩阵乘法的分配律。先定义 3 个随机矩阵。

```
>> A = rand(4,3)
A =
    0.9501    0.8913    0.0214
    0.2311    0.7621    0.4447
    0.6068    0.4565    0.6154
    0.4860    0.0185    0.7919
>> B = rand(4,3)
B =
    0.9218    0.9355    0.0579
    0.7382    0.9169    0.3529
    [illegible]    0.4103    [illegible]
    0.4057    [illegible]    [illegible]
>> C = rand(3,2)
C =
    0.1389    [illegible]
    [illegible]    0.2722
    [illegible]    [illegible]
```

然后验证乘法的分配律，即

```
>> (A + B) * C
ans =
    0.8051    1.0023
    0.6976    1.2009
    0.5684    0.9928
    0.4682    0.9461
>> A * C + B * C
ans =
    0.8051    1.0023
    0.6976    1.2009
    0.5684    0.9928
    0.4682    0.9461
```

除了交换律不满足之外，在矩阵等式中不能用约去运算。即如只 $\boldsymbol{AB} = \boldsymbol{CB}$，不能得出 $\boldsymbol{A} = \boldsymbol{C}$，这可以用下列特例来证明：

$$\begin{bmatrix} 2 & 1 \\ 4 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \end{bmatrix} \tag{2-27}$$

但是

$$\begin{bmatrix} 2 & 1 \\ 4 & 0 \end{bmatrix} \neq \begin{bmatrix} 4 & 0 \\ 0 & 2 \end{bmatrix} \tag{2-28}$$

然而，必须指出的是，如果对于所有的 $\boldsymbol{B}$，$\boldsymbol{AB} = \boldsymbol{CB}$ 都能成立，那么就有 $\boldsymbol{A} = \boldsymbol{C}$，因为此时我们只要令 $\boldsymbol{B} = \boldsymbol{I}$ 即可。

因此也可以得到，如果 $\boldsymbol{AB} = \boldsymbol{0}$，并不能得出 $\boldsymbol{A} = \boldsymbol{0}$ 或 $\boldsymbol{B} = \boldsymbol{0}$，例如：

$$\boldsymbol{A} = \begin{bmatrix} 2 & 4 \\ -3 & -6 \end{bmatrix},\quad \boldsymbol{B} = \begin{bmatrix} -2 & 4 \\ 1 & -2 \end{bmatrix},\quad \boldsymbol{AB} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2-29}$$

另外，矩阵积的转置有如下的规律：

$$(\boldsymbol{AB})^{\mathrm{T}} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}} \tag{2-30}$$

这可以很方便地从矩阵相乘的定义(2-21)来证明。

如果 $\boldsymbol{A}$ 是对称的，那么 $\boldsymbol{B}^{\mathrm{T}}\boldsymbol{AB}$ 也是对称的，证明如下：

$$(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{AB})^{\mathrm{T}} = (\boldsymbol{AB})^{\mathrm{T}}(\boldsymbol{B}^{\mathrm{T}})^{\mathrm{T}} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{B} = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{AB} \tag{2-31}$$

式(2-31)在有限元公式中经常出现，实际上，一般单元的刚度矩阵和质量矩阵都是这种形式。

5. 逆矩阵

从矩阵的加法和矩阵与常数相乘的运算可以很方便地得出矩阵相减与矩阵相加具有相同的运算法则。但是，我们不能从矩阵相乘来推出矩阵的除法。实际上，到目前为止，还没有矩阵除法的定义。矩阵相除用逆矩阵来代替。

方阵 $\boldsymbol{A}$ 的逆矩阵一般用 $\boldsymbol{A}^{-1}$ 来表示。假设逆矩阵存在，则 $\boldsymbol{A}^{-1}$ 的元素由 $\boldsymbol{AA}^{-1} = \boldsymbol{I}$ 或 $\boldsymbol{A}^{-1}\boldsymbol{A} = \boldsymbol{I}$ 来确定。如果矩阵存在逆矩阵，则它为非奇异矩阵，反之，称它为奇异矩阵。

下面就来计算矩阵积 $\boldsymbol{AB}$ 的逆矩阵，令 $\boldsymbol{G} = (\boldsymbol{AB})^{-1}$，根据定义，则有

$$\boldsymbol{GAB} = \boldsymbol{I} \tag{2-32}$$

等式两边右乘 $\boldsymbol{B}^{-1}$，然后再右乘 $\boldsymbol{A}^{-1}$，则得到

$$\boldsymbol{GA} = \boldsymbol{B}^{-1} \tag{2-33}$$

$$\boldsymbol{G} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1} \tag{2-34}$$

因此

$$(\boldsymbol{AB})^{-1} = \boldsymbol{B}^{-1}\boldsymbol{A}^{-1} \tag{2-35}$$

由此注意到，矩阵积的逆矩阵运算与矩阵积的转置运算有相似的规律。

为了得到一般矩阵的逆矩阵，我们可以求解 n 个 n 阶的线性方程组：

$$AX = I \tag{2-36}$$

但是这种方法效率太低，实际计算中一般采用 Gauss 消去法。同时，也可从计算逆矩阵来求解线性方程组，例如有方程组

$$Ay = c \tag{2-37}$$

这里 A 是 $n \times n$ 的方阵，y 和 c 是 $n \times 1$ 阶的向量，两边左乘 A^{-1}，得到

$$y = A^{-1} c \tag{2-38}$$

但是，用一般的方法求解一个矩阵的逆矩阵比用 Gauss 消去法要费时得多。因此式(2-38)只是一种写法，在实际求解方程(2-37)时一般应采用 Gauss 消去法。

对于一般的程序语言，矩阵的求逆必须编写一段复杂的程序，根据矩阵的特点，选择合适的算法，十分麻烦。而对于 **MATLAB**，则是轻而易举，一个 inv 函数就足够了。例如，

```
>> A = rand(4,4)
A =
    0.0153    0.4660    0.2026    0.0019
    [illegible]    [illegible]    0.6721    [illegible]
    0.4451    0.8462    0.8381    0.8318
    0.9318    0.5252    0.0196    0.5028
[illegible]
[illegible]
    [illegible]
   -5.8783   -5.4852    5.5695    1.1981
    [illegible]
```

上面的矩阵 B 就是 A 的逆矩阵，这里可以通过下面的命令可以验证它们的互逆关系：

```
>> [illegible]
ans =
    1.0000         0         0         0
         0    1.0000         0         0
         0         0    1.0000         0
         0         0         0    1.0000
```

A 和 B 相乘的结果是单位矩阵，因此它们是互逆的。

6. 矩阵分块

把矩阵分成小的子矩阵有助于矩阵的操作和利用特殊形式的矩阵。子矩阵可以从原矩阵中的某些行和列的元素来组成，例如，

$$A=\left[\begin{array}{c:ccc:cc} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} \\ \hdashline a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} \end{array}\right] \tag{2-39}$$

可以把矩阵 A 写成

$$A=\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{bmatrix} \tag{2-40}$$

式中

$$A_{11}=\begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix}, \quad A_{12}=\begin{bmatrix} a_{12} & a_{13} & a_{14} \\ a_{22} & a_{23} & a_{24} \end{bmatrix}, \cdots \tag{2-41}$$

矩阵分块以后，它的加法、减法和乘法的运算法则不变，即可以把子矩阵看作单个元素。当然，子矩阵的加法、减法和乘法必须能够进行。

MATLAB 中分块矩阵是通过冒号来实现的。设 A 是一个 5×5 的矩阵，那么

A(2,:)表示矩阵 A 的第二行；

A(:,3) 表示矩阵 A 的第三列；

A(2:4,:)表示由矩阵 A 的第 2,3 和 4 行元素组成的一个 3×5 的矩阵；

A(:,3:5) 表示由矩阵 A 的第 3,4 和 5 列元素组成的一个 5×3 的矩阵；

A(2:4,3:5) 表示由矩阵 A 的第 2 行到第 4 行，第 3 列到第 5 列元素组成的一个 3×3的矩阵；

A(1:2:5,:)表示由矩阵 A 的第 1,3 和 5 行元素组成的一个 3×5 的矩阵。

例如：

```
[illegible]
A =
    0.7095    [illegible]    0.3784    0.8998    0.1400
    0.4289    0.3028    0.8600    0.8216    0.2897
    0.3046    0.6417    0.8537    0.6449    0.8412
    0.1897    0.1509    0.5936    0.8180    0.5341
    0.1934    0.6979    [illegible]    [illegible]    [illegible]
>> A(2,:)
ans =
    0.4289    0.3028    0.8600    0.8216    0.2897
```

```
>> A(:,3)
ans =
    0.3784
    0.8600
    0.8537
    0.5936
    0.4966
>> A(2:4,:)
ans =
    0.4289    0.3028    0.8600    0.8216    0.2897
    0.3046    0.5417    0.8537    0.6449    0.3412
    0.1897    0.1509    0.5936    0.8180    0.5341
>> A(:,3:5)
ans =
    0.3784    0.8998    0.3420
    0.8600    0.8216    0.2897
    0.8537    0.6449    0.3412
    0.5936    0.8180    0.5341
    0.4966    0.6602    0.7271
>> A(2:4,3:5)
ans =
    0.8600    0.8216    0.2897
    0.8537    0.6449    0.3412
    0.5936    0.8180    0.5341
>> A(1:2:5,:)
ans =
    0.7095    0.6822    0.3784    0.8998    0.3420
    0.3046    0.5417    0.8537    0.6449    0.3412
    0.1934    0.6938    0.4966    0.6602    0.7271
```

可见,冒号在 MATLAB 中的应用十分灵活。

7. 矩阵的迹和行列式

矩阵的迹和行列式只有当矩阵为方阵时才有定义。

n 阶方阵 $\boldsymbol{A}$ 的迹用符号 $\mathrm{tr}(\boldsymbol{A})$ 表示,它是矩阵的对角元素之和,即 $\mathrm{tr}(\boldsymbol{A}) = \sum_{i=1}^{n} a_{ii}$。而 n 阶方阵 $\boldsymbol{A}$ 的行列式用 $\det(\boldsymbol{A})$ 表示,它有下式的递归定义:

$$\det(A) = \sum_{j=1}^{n} (-1)^{1+j} a_{1j} \det(A_{1j}) \tag{2-42}$$

式中 A_{1j} 是 $(n-1)$ 阶方阵，它是将矩阵 A 的第 1 行和第 j 列划去后得到的。多个矩阵相乘后的行列式有以下公式

$$\det(BC\cdots F) = \det(B)\det(C)\cdots\det(F) \tag{2-43}$$

式(2-43)的证明是非常冗长的，因此这里没有给出。利用上式和矩阵分解可简化矩阵行列式的计算，也就是说可以把一个矩阵 A 写成 $A = LDL^{T}$，其中矩阵 L 的对角元素全部为 1，且对角元一方的元素全部为零，而 D 是一个对角阵，即除对角元外的元素全为零。则有

$$\det(A) = \det(L)\det(D)\det(L^{T}) \tag{2-44}$$

因为 $\det(L) = \det(L^{T}) = 1$，则有

$$\det(A) = \prod_{i=1}^{n} d_{ii} \tag{2-45}$$

MATLAB 中用两个函数 trace 和 det 分别求取矩阵的迹和行列式值，例如，

```
>> A=[3 [illegible] 6 3
      0 2 -2 3
      0 0 -2 3
      0 0 0 [illegible]]
A =
    3  [illegible]  6  3
    0  2  -2  3
    0  0  -2  3
    0  0   0  [illegible]
>> trace(A)
ans =
    7
>> det(A)
ans =
```

[illegible]

2.2 MATLAB 语言程序设计初步

MATLAB 语言不仅可在命令窗口中直接输入命令，得出结果，还具有编程的功能。要实现一个计算机语言，它必须有两个基本的要素，一个是条件分

支，另一个是循环。MATLAB中实现条件分支有两种方法，分别是 if 和 switch 语句，现分别介绍如下。这里可以利用 MATLAB 提供的在线帮助功能，看看它们是如何使用的。

```
>> help if

IF statement condition.
  The general form of the IF statement is

    IF expression
      statements
    ELSEIF expression
      statements
    ELSE
      statements
    END

  The statements are executed if the real part of the expression has all
  non-zero elements. The ELSE and ELSEIF parts are optional.
  Zero or more ELSEIF parts can be used as well as nested IF's.
  The expression is usually of the form expr rop expr where
  rop is ==, <, >, <=, >=, or ~=.

  Example
    if I == J
      A(I,J) = 2;
    elseif abs(I-J) == 1
      A(I,J) = -1;
    else
      A(I,J) = 0;
    end
```

可见，MATLAB的语法跟C语言非常相似，如果熟悉C语言，学习 MATLAB就能事半功倍。这里需要注意的时，在 MATLAB 的"帮助"文档中，关键词都是以大写的形式出现的，但是在实际使用时，必须是小写的形式。下面是关于 switch 的用法。

```
>> help switch
SWITCH Switch among several cases based on expression.
```

The general form of the SWITCH statement is:

```
SWITCH switch_expr
  CASE case_expr,
    statement, ..., statement
  CASE {case_expr1, case_expr2, case_expr3,...}
    statement, ..., statement
  ...
  OTHERWISE,
    statement, ..., statement
END
```

The statements following the first CASE where the switch_expr matches the case_expr are executed. When the case expression is a cell array (as in the second case above), the case_expr matches if any of the elements of the cell array match the switch expression. If none of the case expressions match the switch expression then the OTHERWISE case is executed (if it exists). Only one CASE is executed and execution resumes with the statement after the END.

The switch_expr can be a scalar or a string. A scalar switch_expr matches a case_expr if switch_expr==case_expr. A string switch_expr matches a case_expr if strcmp(switch_expr,case_expr) returns 1 (true).

Only the statements between the matching CASE and the next CASE, OTHERWISE, or END are executed. Unlike C, the SWITCH statement does not fall through (so BREAKs are unnecessary).

Example:

To execute a certain block of code based on what the string, METHOD, is set to,

```
    method = 'Bilinear';

switch lower(METHOD)
```

```
case {'linear','bilinear'}
    disp('Method is linear')
case 'cubic'
    disp('Method is cubic')
case 'nearest'
    disp('Method is nearest')
otherwise
    disp('Unknown method.')
end

Method is linear
```

这里的"帮助"文档中特别强调了 MATLAB 的 switch 语句与 C 语言中的 switch 的不同，也就是在 MATLAB 的每一个 case 分支后面不需要用 break 来强制退出 switch 语句。实现循环的方法也有两种，分别是 for 和 while。MATLAB 对它们的说明如下：

```
>> help for

FOR    Repeat statements a specific number of times.
  The general form of a FOR statement is:

   FOR variable = expr, statement, ..., statement END

  The columns of the expression are stored one at a time in
  the variable and then the following statements, up to the
  END, are executed. The expression is often of the form X:Y,
  in which case its columns are simply scalars. Some examples
  (assume N has already been assigned a value).

      FOR I = 1:N,
          FOR J = 1:N,
              A(I,J) = 1/(I+J-1);
          END
      END

  FOR S = 1.0: -0.1: 0.0, END steps S with increments of -0.1
  FOR E = EYE(N), ... END  sets E to the unit N-vectors.
```

```
Long loops are more memory efficient when the colon expression appears
in the [illegible].

The BREAK statement [illegible].
以及 while [illegible]

>> help while

WHILE  Repeat statements an indefinite number of times.
    The general form of a WHILE statement is:

       WHILE expression
         statements
       END

    The statements are executed while the real part of the expression has
    all non-zero elements. The expression is usually the result of expr
    rop expr where rop is ==, <, >, <=, >=, or ~=.

    The BREAK statement can be used to terminate the loop prematurely.

    For example (assuming A already defined):

          E = 0*A; F = E + eye(size(E)); N = 1;
          while norm(E+F-E,1) > 0,
            E = E + F;
            F = A*F/N;
            N = N + 1;
          end
```

用 MATLAB 语言编写的程序都应该保存在扩展名为.m 的文件中，称为 m 文件。这种文件的第一行一般应该是：

```
function myfun
```

这个语句的意思是定义一个函数，相当于 FORTRAN 程序中的子程序和 C 语言的函数。如果没有这个语句，则这种 m 文件可称为脚本(script)文件，功能是一样的，区别在于程序中的变量全部是全局的。

2.3 弹性力学的控制方程和最小势能原理

根据弹性理论，弹性体内的应力、应变和位移要满足平衡方程，几何方程和本构关系以及相应的边界条件，下面是矩阵形式的控制方程[21]：

平衡方程： $$L\boldsymbol{\sigma} + \boldsymbol{p}_v = 0 \tag{2-46}$$

几何方程： $$\boldsymbol{\varepsilon} = \boldsymbol{L}^{T} \boldsymbol{u} \tag{2-47}$$

本构关系： $$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\varepsilon} \tag{2-48}$$

以及边界条件

在指定力的边界 B_2 上： $$\boldsymbol{n}\boldsymbol{\sigma} = \boldsymbol{p}_s \tag{2-49}$$

在指定位移的边界 B_1 上： $$\boldsymbol{u} = \bar{\boldsymbol{u}} \tag{2-50}$$

式中 $\boldsymbol{\sigma}$、$\boldsymbol{\varepsilon}$ 和 $\boldsymbol{u}$ 分别是应力，应变和位移，$\boldsymbol{p}_v$ 是体积力，$\boldsymbol{L}$ 是微分算子矩阵，$\boldsymbol{D}$ 是弹性矩阵，表示应力应变关系，而 $\boldsymbol{p}_s$ 和 $\bar{\boldsymbol{u}}$ 是边界上已知的外力和位移，$\boldsymbol{n}$ 是边界外法线方向矩阵。在直角坐标系 xyz 下，可以给出它们的表达式分别为：

$$\boldsymbol{\sigma} = [\sigma_x \quad \sigma_y \quad \sigma_z \quad \tau_{yz} \quad \tau_{xz} \quad \tau_{xy}]^{T}$$

$$\boldsymbol{\varepsilon} = [\varepsilon_x \quad \varepsilon_y \quad \varepsilon_z \quad \gamma_{yz} \quad \gamma_{xz} \quad \gamma_{xy}]^{T} \tag{2-52}$$

$$\boldsymbol{u} = [u \quad v \quad w]^{T} \tag{2-53}$$

$$\boldsymbol{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} \\ 0 & 0 & \frac{\partial}{\partial z} & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} & 0 \end{bmatrix} \tag{2-54}$$

$$\boldsymbol{n} = \begin{bmatrix} n_x & 0 & 0 & 0 & n_z & n_y \\ 0 & n_y & 0 & n_z & 0 & n_x \\ 0 & 0 & n_z & n_y & n_x & 0 \end{bmatrix} \tag{2-55}$$

式中 $\sigma_x, \sigma_y, \cdots, \tau_{xy}$ 和 $\varepsilon_x, \varepsilon_y, \cdots, \gamma_{xy}$ 分别是应力和应变分量，u，v 和 w 是3个位移分量；n_x，n_y 和 n_z 是边界外法线的方向余弦。对于任意各向异性材料，弹性矩阵为

$$
\boldsymbol{D}=\begin{bmatrix}
c_{11} & c_{12} & c_{13} & c_{14} & c_{15} & c_{16} \\
 & c_{22} & c_{23} & c_{24} & c_{25} & c_{26} \\
 & & c_{33} & c_{34} & c_{35} & c_{36} \\
 & \text{对} & & c_{44} & c_{45} & c_{46} \\
 & & \text{称} & & c_{55} & c_{56} \\
 & & & & & c_{66}
\end{bmatrix} \tag{2-56}
$$

当材料有弹性对称轴时，弹性矩阵中的独立变量数目将减少。例如对于各向同性材料，只有两个是独立的，一般用弹性模量 E 和泊松比 ν 来表示，此时，矩阵 $\boldsymbol{D}$ 可简化为

$$
\boldsymbol{D}=\frac{E(1-\nu)}{(1+\nu)(1-2\nu)}\begin{bmatrix}
1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\
 & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\
 & & 1 & 0 & 0 & 0 \\
 & \text{对} & & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\
 & & \text{称} & & \frac{1-2\nu}{2(1-\nu)} & 0 \\
 & & & & & \frac{1-2\nu}{2(1-\nu)}
\end{bmatrix} \tag{2-57}
$$

我们把满足所有控制方程和边界条件式（2-46）~式（2-50）的位移称为真实位移，而把只满足边界条件的位移称为可能位移，那么最小势能原理就可以表述为在所有的可能位移中，真实位移使系统的总势能取最小值。

这里整个系统的势能包括两部分：一部分是弹性体的应变能 π_1，它的算式是

$$
\pi_1=\frac{1}{2}\iiint_{\Omega}\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\sigma}\mathrm{d}\Omega \tag{2-58}
$$

式中，积分区域 $\boldsymbol{\Omega}$ 即为弹性体在空间所占的体积。另一部分是外荷载的势能 π_2，它的算式是

$$
\pi_2=-\iiint_{\Omega}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{p}_v\mathrm{d}\Omega-\iint_{B_\sigma}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{p}_s\mathrm{d}B \tag{2-59}
$$

式中边界 B_2 是指定力的边界，p_s 是作用在上面的分布面力；p_v 是体积力。所以整个系统的势能 π 是

$$\pi = \pi_1 + \pi_2 \tag{2-60}$$

当把应力 σ 和应变 ε 都用位移 u 表示后，系统的势能 π 可写为

$$\pi = \frac{1}{2}\iiint_{\Omega} u^{T} LDL^{T} u \mathrm{d}\Omega - \iiint_{\Omega} u^{T} p_v \mathrm{d}\Omega - \iint_{B_2} u^{T} p_s \mathrm{d}B \tag{2-61}$$

它是依赖自变函数 u 而变的量，因此就称它为是自变函数 u 的泛函。

最小势能原理指出，真实位移 u 将使系统的势能 π 取极小值。根据变分原理，欲使 π 取极小值，必须有泛函 π 的变分为零，即

$$\delta\pi = 0 \tag{2-62}$$

求泛函的极值是变分法的主要工作。早期的变分法主要是把泛函的极值问题转化为微分方程，而有限元中的变分法则是要把微分方程转化为泛函的极值问题。这里只是十分扼要地介绍了变分法和最小势能原理，有兴趣的同学可参考相应的专著[22,23]。

2.4 有限元法分析过程的概述

为便于学习和理解以后各章的内容，现在把该法的分析过程概述如下。

1. 结构的离散化

结构的离散化是有限单元法分析的第一步，它是有限单元法的基本概念。所谓离散化，简单地说就是将要分析的结构物分割成有限个单元体，并在单元体的指定点设置结点，使相邻单元的有关参数具有一定的连续性，并构成一个单元的集合体，用它代替原来的结构。如果分析的对象是桁架，那么这种划分十分明显，可以取每根杆件作为一个单元，因为桁架本来就是由杆件组成的。但是，如果分析的对象是连续体，那么为了有效地逼近实际的连续体，就需要考虑选择单元的形状和分割方案以及确定单元和结点的数目等问题。有关这方面的内容，将在以后的章节中逐一介绍。

2. 选择位移模式

在完成结构的离散之后，就可以对典型单元进行特性分析。此时，为了能用结点位移表示单元体的位移、应变和应力，在分析连续体问题时，必须对单元中位移的分布作出一定的假设，也就是假定位移是坐标的某种简单的函数，这种函数称为位移模式或插值函数。

选择适当的位移函数是有限单元法分析中的关键。通常选择多项式作为

位移模式。其原因是因为多项式的数学运算（微分和积分）比较方便，并且由于所有光滑函数的局部，都可以用多项式逼近。至于多项式的项数和阶次的选择，则要考虑到单元的自由度和解的收敛性。一般来说，多项式的项数应等于单元的自由度数，它的阶次应包含常数项和线性项等。这里所谓单元的自由度是指单元结点独立位移的个数。

根据所选定的位移模式，就可以导出用结点位移表示单元内任一点位移的关系式，其矩阵形式是

$$\boldsymbol{u} = \boldsymbol{N}\boldsymbol{\delta}^e \tag{2-63}$$

式中，$\boldsymbol{u}$ 是单元内任一点的位移向量，$\boldsymbol{\delta}^e$ 为单元的结点位移向量，$\boldsymbol{N}$ 称为形函数矩阵，它的元素是位置坐标的函数。单元内的应变和应力分别为

$$\boldsymbol{\varepsilon} = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{N}\boldsymbol{\delta}^e = \boldsymbol{B}\boldsymbol{\delta}^e \tag{2-64}$$

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\varepsilon} = \boldsymbol{D}\boldsymbol{B}\boldsymbol{\delta}^e = \boldsymbol{S}\boldsymbol{\delta}^e \tag{2-65}$$

式中，矩阵 $\boldsymbol{B}$ 和 $\boldsymbol{S}$ 分别称为应变矩阵和应力矩阵。

在此，顺便指出：有限单元法比起经典的近似法具有明显的优越性。例如，在经典的 Ritz 法中，要求选取一个函数来近似地描述整个求解区域中的位移，并必须满足边界条件；而在有限单元法中则采用分块近似，只需要对一个单元选择一个近似函数。此时，不必考虑位移边界条件，只需考虑单元之间位移的连续性就可以了。这样做当然比起在整个区域内选取一个连续函数要简单得多，特别是对于复杂的几何形状或者材料特性、作用荷载有突变的结构，采用分段函数，就显得更是合理和适宜了。

3. 分析单元的力学特性

位移模式选定以后，就可以进行单元的力学特性的分析，它的主要目的是得到作用于单元上的结点力和结点位移之间的关系式，下面采用 2.3 节中能量原理来推导单元的平衡方程。把式(2-63)代入式(2-61)，得到

$$\pi = \frac{1}{2}\boldsymbol{\delta}^{e\mathrm{T}}\left(\iiint_{\Omega}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}\,\mathrm{d}\Omega\right)\boldsymbol{\delta}^e - \boldsymbol{\delta}^{e\mathrm{T}}\left(\iiint_{\Omega}\boldsymbol{N}^{\mathrm{T}}\boldsymbol{p}_v\,\mathrm{d}\Omega\right) - \boldsymbol{\delta}^{e\mathrm{T}}\left(\iint_{B_2}\boldsymbol{N}^{\mathrm{T}}\boldsymbol{p}_s\,\mathrm{d}B\right) \tag{2-66}$$

根据最小势能原理，式(2-66)中势能的变分为零，即

$$\delta\pi = \frac{\partial\pi}{\partial\boldsymbol{\delta}^e}\delta\boldsymbol{\delta}^e = 0 \tag{2-67}$$

由于结点位移向量 $\boldsymbol{\delta}^e$ 的任意性，则有

$$\frac{\partial \pi}{\partial \boldsymbol{\delta}^e} = \left(\iiint_{\Omega} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}\mathrm{d}\Omega\right)\boldsymbol{\delta}^e - \iiint_{\Omega} \boldsymbol{N}^{\mathrm{T}}\boldsymbol{p}_v \,\mathrm{d}\Omega - \iint_{B_2} \boldsymbol{N}^{\mathrm{T}}\boldsymbol{p}_s\mathrm{d}B = 0 \tag{2-68}$$

上式可以写成

$$\boldsymbol{K}^e \boldsymbol{\delta}^e = \boldsymbol{f}^e \tag{2-69}$$

式中，$\boldsymbol{K}^e$ 称为单元刚度矩阵，$\boldsymbol{f}^e$ 为等效结点力，它们分别为

$$\boldsymbol{K}^e = \iiint_{\Omega} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}\mathrm{d}\Omega \tag{2-70}$$

$$\boldsymbol{f}^e = \boldsymbol{f}_v + \boldsymbol{f}_s \tag{2-71}$$

式中体积力的等效结点力 $\boldsymbol{f}_v$ 和面力的等效结点力 $\boldsymbol{f}_s$ 分别为

$$\boldsymbol{f}_v = \iiint_{\Omega} \boldsymbol{N}^{\mathrm{T}}\boldsymbol{p}_v \,\mathrm{d}\Omega, \quad \boldsymbol{f}_s = \iint_{B_2} \boldsymbol{N}^{\mathrm{T}}\boldsymbol{p}_s\mathrm{d}B \tag{2-72}$$

4. 集合所有单元的平衡方程，建立整个结构的平衡方程

这个集合过程包括有两个方面的内容：一是将各个单元的刚度矩阵集合成整个结构的整体刚度矩阵；二是将作用于各单元的等效结点力向量集合成总的荷载向量。最常用的集合刚度矩阵的方法是直接刚度法。一般来说，集合所依据的理由是要求所有相邻的单元在公共结点上的位移相等。于是得到用整体刚度矩阵 $\boldsymbol{K}$、荷载向量 $\boldsymbol{f}$ 和整个结构的结点位移向量 $\boldsymbol{\delta}$ 表示的整个结构的平衡方程

$$\boldsymbol{K}\boldsymbol{\delta} = \boldsymbol{f} \tag{2-73}$$

5. 求解未知结点位移和计算单元内力或应力

对方程(2-73)作适当修改以考虑结构的边界条件后解出结点位移 $\boldsymbol{\delta}$，然后再利用单元特性计算单元应力或内力。

参考文献

[1] Munkres J. R.. Elementary Linear Algebra. Addison-Wesley, 1964.

[2] Roman S.. Advanced Linear Algebra. Springer-Verlag, 1992.

[3] 豪斯霍尔德，孙家昶，等译. 数值分析中的矩阵论. 北京：科学出版社，1986.

[4] Etter D.. Engineering Problem Solving with MATLAB. 2nd Edition. Prentice Hall, 1996.

[5] Part-Enander E., Sjoberg A. The MATLAB 5 Handbook. Addision-Wesley, 1999.

[6] Biram A., Breiner M.. MATLAB 5 for Engineering. Addision-Wesley, 1999.

[7] Higham D., Higham N.. MATLAB Guide. SIAM, 2000.

[8] Hanselman D. , Littlefield B. . Mastering MATLAB 6. Prentice Hall,2000.

[9] Palm W. . Introduction to MATLAB 6 for Engineerings. McGraw-Hill,2000.

[10] Chen K. , Giblin P. , Irving A. . Mathematical Explorations with MATLAB. Cambridge University Press,1999.

[11] Polking J. , Arnold D. . Ordinary Differential Equations Using MATLAB. 2nd Edition. Prentice Hall,1999.

[12] Fausett L. . Applied Numerical Analysis Using MATLAB. Prentice Hall,1999.

[13] Cavers I. . An Introductory Guide to MATLAB. 2nd Edition,[1998]. http://www. cs. ubc. ca/spider/cavers/MatlabGuide/guide. html.

[14] Gockenback M. . A Practical Introduction to MATLAB: Updated for MATLAB 5. http://www. math. mtu. edu/ ~ msgocken/intro/intro. html.

[15] Huber T. . MATLAB Tutorials: Envision-It! Workshop. [1997]. http://physics. gac. edu/ ~ huber/envision/tutor2.

[16] Maneval J. Helpful Information for Using MATLAB. http://www. facstaff. bucknell. edu/ maneval/help211/helpmain. html.

[17] Recktenwald G,. MATLAB Hypertext Reference. [1996]. http://www. me. pdx. edu/ ~ gerry/MATLAB.

[18] University of Texas. MATLAB Online Reference Documentation. http://www. utexas. edu/ cc/math/Matlab/Manual/ReferenceTOC. html.

[19] Southern Illinois University at Carbondale. MATLAB Tutorials. http://www. math. siu. edu/matlab/tutorials. html.

[20] MATLAB Educational Sites. http://www. eece. maine. edu/mm/matweb. html.

[21] 王勖成. 有限单元法. 北京:清华大学出版社,2003.

[22] 胡海昌. 弹性力学的变分原理及其应用. 北京:科学出版社,1981.

[23] 鹫津久一郎. 弹性和塑性力学中的变分法. 北京:科学出版社,1984.

第3章 杆件系统的有限单元法

有限元法与结构力学中刚架计算的矩阵位移法有密切关系，因此本节从杆件系统的有限元法入手，来讲述有限元法的基本概念和过程。杆件系统的有限单元法和结构力学中经典的位移法比较，并没有很大的区别。只是在有限单元法中，基本结构的选取有所不同，凡是杆系的交叉点、边界点、集中力作用点都应列为结点，而结点之间的杆件均作为单元。也就是说，用单元代替了经典位移法中的基本结构[1]。

在杆件系统中，根据单元受力的特点可以把它们分成两大类：杆和梁。为了以后描述的方便，把两端铰接，只受轴向力的基本结构称为杆单元，而受轴向力和弯矩、扭矩、剪力共同作用的基本结构称为梁单元。根据其截面的一个主轴与它所承受的荷载是否在同一平面内，杆件单元可分成平面单元或空间单元。

3.1 杆单元

设有一任意的杆单元如图 3-1 所示，i 和 j 为单元的两个结点，x 为该单元的局部坐标，其原点设在单元的 i 结点。设两个结点在 x 方向的位移为 u_i 和 u_j，它们的正方向如图 3-1 所示，与它们相应的结点力 F_{Ni} 和 F_{Nj} 正方向也示于图中。



图 3-1 杆单元

根据材料力学[2]的有关知识，则可以立刻写出杆单元的结点位移与结点力之间的关系为

$$F_{Ni} = \frac{EA}{l}(u_i - u_j) \qquad F_{Nj} = \frac{EA}{l}(u_j - u_i) \tag{3-1}$$

式中，E 为材料的弹性模量，A 和 l 分别是杆单元的横截面面积和长度。为了方便，把上式写成矩阵形式

$$\boldsymbol{f}^e = \begin{Bmatrix} F_{Ni} \\ F_{Nj} \end{Bmatrix} = \frac{EA}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} u_i \\ u_j \end{Bmatrix} = \boldsymbol{K}^e\boldsymbol{\delta}^e \tag{3-2}$$

式中

$$\boldsymbol{K}^e = \frac{EA}{l}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{3-3}$$

为杆单元的单元刚度矩阵，我们可以很容易发现它是对称的，也是奇异的。

上面是利用材料力学的知识直接导出了杆单元的刚度矩阵，但是对于一般的单元，则通常采用能量变分原理。下面就来详细说明。

首先要把单元内的位移用结点位移来表示，为此，就要先假设一个位移模式。由于每个结点只有一个轴向位移，即一个单元共有两个自由度，因此可以假设单元内的位移模式为

$$u = a_0 + a_1 x = \begin{bmatrix} 1 & x \end{bmatrix}\begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \boldsymbol{Ha} \tag{3-4}$$

把单元结点的坐标和位移代入，就能得到关于待定参数 $\boldsymbol{a}$ 的线性方程组

$$\begin{bmatrix} 1 & 0 \\ 1 & l \end{bmatrix}\begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} u_i \\ u_j \end{Bmatrix} \tag{3-5}$$

解之即能得到

$$\boldsymbol{a} = \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & l \end{bmatrix}^{-1}\begin{Bmatrix} u_i \\ u_j \end{Bmatrix} \tag{3-6}$$

把式(3-6)代入式(3-4)，就得到了用结点位移表示的单元位移

$$u = \begin{bmatrix} 1 & x \end{bmatrix}\begin{bmatrix} 1 & 0 \\ 1 & l \end{bmatrix}^{-1}\begin{Bmatrix} u_i \\ u_j \end{Bmatrix} = \boldsymbol{N\delta}^e \tag{3-7}$$

式中形函数 $\boldsymbol{N}$ 和结点位移列阵 $\boldsymbol{\delta}^e$ 为

$$\boldsymbol{N} = \begin{bmatrix} 1 - \frac{x}{l} & \frac{x}{l} \end{bmatrix}, \quad \boldsymbol{\delta}^e = \begin{Bmatrix} u_i \\ u_j \end{Bmatrix} \tag{3-8}$$

有了单元内位移的表达式(3-7)，单元内的应变和应力可以很容易得到

$$\varepsilon_x = \frac{\mathrm{d}u}{\mathrm{d}x} = \left[-\frac{1}{l} \quad \frac{1}{l} \right] \begin{Bmatrix} u_i \\ u_j \end{Bmatrix} = \boldsymbol{B}\boldsymbol{\delta}^e \tag{3-9}$$

$$\sigma_x = E\varepsilon_x = E\boldsymbol{B}\boldsymbol{\delta}^e = \boldsymbol{S}\boldsymbol{\delta}^e \tag{3-10}$$

式中应变矩阵 $\boldsymbol{B}$ 为

$$\boldsymbol{B} = \left[-\frac{1}{l} \quad \frac{1}{l} \right] \tag{3-11}$$

把式(3-11)代入式(2-70)，并注意到此时弹性矩阵 $\boldsymbol{D} = E$，就得到杆单元的刚度矩阵为

$$\boldsymbol{K}^e = \iiint_{\Omega} \boldsymbol{B}^{\mathrm{T}} E \boldsymbol{B} \mathrm{d}\Omega = EAl\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B} \tag{3-12}$$

只要把式(3-11)代入式(3-12)即能得到跟式(3-3)一致的单元刚度矩阵。

3.2 平面梁单元

设有任意平面梁单元，如图3-2所示，其结点为 i 和 j。利用右手坐标系，使 x 轴与梁轴重合，而 y 轴和 z 轴为梁截面的主惯性轴方向。由于荷载都在同一平面内，梁单元处于轴向拉压和平面弯曲的组合变形状态，在结点 i 和 j 上所受的结点力为轴力、剪力和弯矩，它们分别是 F_{Ni}、F_{Qi}、M_i 和 F_{Nj}、F_{Qj}、M_j；与之对应的结点位移分别是 u_i、v_i、θ_i 和 u_j、v_j、θ_j。这里 θ_i 和 θ_j 是结点 i 和 j 的转角位移，它们是挠度 v 对 x 的导数。图3-2所示的结点力和结点位移的方向均取为正方向。

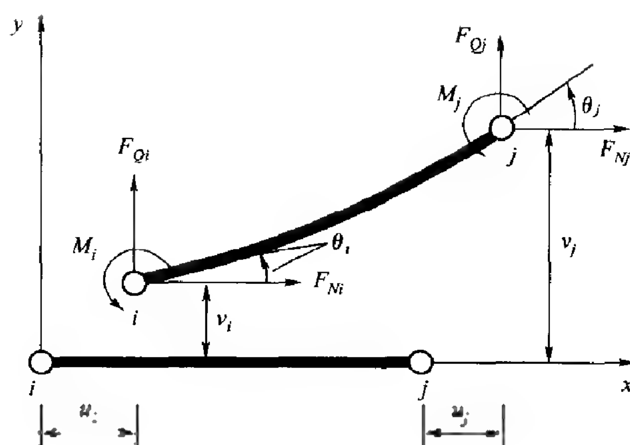


图3-2 平面梁单元

这里同样采用最小势能原理来推导平面梁单元的单元刚度矩阵。轴向位

移 u 的位移模式跟杆单元一样,取 x 的线性函数,而挠度 v 则用三次多项式表示。于是有

$$v = b_0 + b_1 x + b_2 x^2 + b_3 x^3 \tag{3-13}$$

若写成矩阵形式，则有

$$u = \boldsymbol{H}_1 \boldsymbol{a} \qquad v = \boldsymbol{H}_2 \boldsymbol{b} \tag{3-14}$$

式中

$$\boldsymbol{H}_1 = \begin{bmatrix} 1 & x \end{bmatrix} \qquad \boldsymbol{H}_2 = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \tag{3-15}$$

$$\boldsymbol{a} = \begin{bmatrix} a_0 & a_1 \end{bmatrix}^T \qquad \boldsymbol{b} = \begin{bmatrix} b_0 & b_1 & b_2 & b_3 \end{bmatrix}^T \tag{3-16}$$

在式(3-13)中的参数 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 是位移模式的待定参数，它们可以由结点位移来表示。

将轴向结点位移、结点挠度和转角记为

$$\boldsymbol{\delta}_u = \begin{bmatrix} u_i & u_j \end{bmatrix}^T \qquad \boldsymbol{\delta}_v = \begin{bmatrix} v_i & \theta_i & v_j & \theta_j \end{bmatrix}^T \tag{3-17}$$

于是,由式(3-13)得到

$$\boldsymbol{\delta}_u = \boldsymbol{A}_u \boldsymbol{a} \qquad \boldsymbol{\delta}_v = \boldsymbol{A}_v \boldsymbol{b} \tag{3-18}$$

式中

$$\boldsymbol{A}_u = \begin{bmatrix} 1 & 0 \\ 1 & l \end{bmatrix} \qquad \boldsymbol{A}_v = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & l & l^2 & l^3 \\ 0 & 1 & 2l & 3l^2 \end{bmatrix} \tag{3-19}$$

由式(3-18)可以得到参数

$$\boldsymbol{a} = \boldsymbol{A}_u^{-1} \boldsymbol{\delta}_u \qquad \boldsymbol{b} = \boldsymbol{A}_v^{-1} \boldsymbol{\delta}_v \tag{3-20}$$

其中,$\boldsymbol{A}_u^{-1}$ 和 $\boldsymbol{A}_v^{-1}$ 分别是矩阵 $\boldsymbol{A}_u$和 $\boldsymbol{A}_v$ 的逆矩阵,它们是

$$\boldsymbol{A}_u^{-1} = \begin{bmatrix} 1 & 0 \\ -1/l & 1/l \end{bmatrix} \quad \boldsymbol{A}_v^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3/l^2 & -2/l & 3/l^2 & -1/l \\ 2/l^3 & 1/l^2 & -2/l^3 & 1/l^2 \end{bmatrix} \tag{3-21}$$

MATLAB不仅可以进行数值运算，也能进行符号运算。如式（3-20）中的矩阵 $\boldsymbol{A}_u$ 和 $\boldsymbol{A}_v$ 的求逆运算，则可以在 MATLAB 的命令窗口下输入:

```
>> syms L
>> Au=[1 0
      0 L];
>> Av=[1 0 0 0
      0 1 0 0
      1 L L^2 L^3
      0 1 2*L 3*L^2];
```

符号“>>”是 MATLAB 的命令提示符,表示前一命令已经完成,可以输入新的命令。第一句是定义符号变量 L,然后定义两个矩阵 Au 和 Av。然后再输入下面求逆的命令:

```
>> inv(Au)
ans =
   [  1,   0]
   [  0, 1/L]
>> inv(Av)
ans =
   [      1,     0,      0,     0]
   [      0,     1,      0,     0]
   [ -3/L^2,-2/L,  3/L^2,  -1/L]
   [  2/L^3, 1/L^2, -2/L^3, 1/L^2]
```

这样就利用 MATLAB 的符号运算功能得到了矩阵 $\boldsymbol{A}_u$ 和 $\boldsymbol{A}_v$ 的逆矩阵了。

若将式(3-20)代入式(3-13),于是得到用结点位移表示的位移模式。它的矩阵形式是

$$u = \boldsymbol{H}_1 \boldsymbol{A}_u^{-1} \boldsymbol{\delta}_u \qquad v = \boldsymbol{H}_2 \boldsymbol{A}_v^{-1} \boldsymbol{\delta}_v \tag{3-22}$$

上面两式或可写成下列形式

$$u = \boldsymbol{N}_u \boldsymbol{\delta}_u \qquad v = \boldsymbol{N}_v \boldsymbol{\delta}_v \tag{3-23}$$

式中

$$\boldsymbol{N}_u = \boldsymbol{H}_1 \boldsymbol{A}_u^{-1} \qquad \boldsymbol{N}_v = \boldsymbol{H}_2 \boldsymbol{A}_v^{-1} \tag{3-24}$$

便是位移的形函数矩阵。

综合公式(3-23),并将结点 i 和 j 的位移列阵表示为

$$\boldsymbol{\delta}_i = [u_i \quad v_i \quad \theta_i]^{\mathrm{T}} \qquad \boldsymbol{\delta}_j = [u_j \quad v_j \quad \theta_j]^{\mathrm{T}} \tag{3-25}$$

以及记

$$\boldsymbol{\delta}^e = [\boldsymbol{\delta}_i^{\mathrm{T}} \quad \boldsymbol{\delta}_j^{\mathrm{T}}]^{\mathrm{T}} \tag{3-26}$$

另外位移模式的表达式也可以改写成矩阵形式

$$\boldsymbol{u} = \begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} \boldsymbol{H}_u \\ \boldsymbol{H}_v \end{bmatrix} \boldsymbol{A}^{-1} \boldsymbol{\delta}^e = \boldsymbol{N} \boldsymbol{\delta}^e \tag{3-27}$$

式中

$$\begin{aligned} \boldsymbol{H}_u &= [1 \quad 0 \quad 0 \quad x \quad 0 \quad 0] \\ \boldsymbol{H}_v &= [0 \quad 1 \quad x \quad 0 \quad x^2 \quad x^3] \end{aligned} \tag{3-28}$$

而

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & l & 0 & 0 \\ 0 & 1 & l & 0 & l^2 & l^3 \\ 0 & 0 & 1 & 0 & 2l & 3l^2 \end{bmatrix} \tag{3-29}$$

梁单元受到拉压和弯曲变形后，它的线应变可以分成两部分：拉压应变 $\boldsymbol{\varepsilon}_0$ 和弯曲应变 $\boldsymbol{\varepsilon}_b$。若略去剪切变形，则有

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_0 \\ \varepsilon_b \end{Bmatrix} = \begin{Bmatrix} \dfrac{\mathrm{d}u}{\mathrm{d}x} \\ -y\dfrac{\mathrm{d}^2 v}{\mathrm{d}x^2} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{H}'_u \\ -y\boldsymbol{H}''_v \end{bmatrix} \boldsymbol{A}^{-1} \boldsymbol{\delta}^e \tag{3-30}$$

上式中的一撇表示对 x 的导数，它可进一步写成

$$\boldsymbol{\varepsilon} = \boldsymbol{B}\boldsymbol{\delta}^e \tag{3-31}$$

式中

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{H}'_u \\ -y\boldsymbol{H}''_v \end{bmatrix} \boldsymbol{A}^{-1} \tag{3-32}$$

而

$$\begin{aligned} \boldsymbol{H}'_u &= [0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0] \\ \boldsymbol{H}''_v &= [0 \quad 0 \quad 0 \quad 0 \quad 2 \quad 6x] \end{aligned} \tag{3-33}$$

由 Hook 定理，就可得到结点位移表示单元中应力的表达式

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_0 \\ \sigma_b \end{Bmatrix} = E\begin{Bmatrix} \varepsilon_0 \\ \varepsilon_b \end{Bmatrix} = E\boldsymbol{B}\boldsymbol{\delta}^e \tag{3-34}$$

再次应用式(2-70)，并进行一系列的积分运算，可以得出单元刚度矩阵的显式如下：

$$
\boldsymbol{K}^e = E\iint_A \mathrm{d}A\int_0^l \boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}\mathrm{d}x
$$

$$
= \left[\begin{array}{ccc:ccc}
EA/l & & & \text{对} & & \text{称} \\
0 & 12EI/l^3 & & & & \\
0 & 6EI/l^2 & 4EI/l & & & \\
\hdashline
-EA/l & 0 & 0 & EA/l & & \\
0 & -12EI/l^3 & -6EI/l^2 & 0 & 12EI/l^3 & \\
0 & 6EI/l^2 & 2EI/l & 0 & -6EI/l^2 & 4EI/l
\end{array}\right] \tag{3-35}
$$

其中，$I=\iint y^2\mathrm{d}A$ 是梁截面对主轴的惯性矩，A 为截面面积。我们同样可以发现梁单元的刚度矩阵 $\boldsymbol{K}^e$ 是对称的，而且是奇异的。

上面刚度矩阵的推导也可以应用 **MATLAB** 的符号运算功能，计算的程序段如下：

```
syms E L x y
A=[1 0 0 0 0 0
   0 1 0 0 0 0
   0 0 1 0 0 0
   1 0 0 L 0 0
   0 1 L 0 L^2 L^3
   0 0 1 0 2*L 3*L^2];
  Hu=[1 0 0 x 0 0];
  Hv=[0 1 x 0 x^2 x^3];
  B=[diff(Hu,x,1); -y*diff(Hv,x,2)]*inv(A);
  Ke=int(E*transpose(B)*B,x,0,L);
```

上述命令使用了符号的微分 `diff` 和积分 `int` 命令，具体使用方法可参考 **MATLAB** 的“帮助”文档或有关书籍。在命令窗口中执行后，得到单元刚度矩阵的表达式如下：

```
Ke =
[   E/L,            0,            0, -E/L,            0,            0]
[     0, 12*E*y^2/L^3,  6*E*y^2/L^2,    0,-12*E*y^2/L^3,  6*E*y^2/L^2]
[     0,  6*E*y^2/L^2,    4*E*y^2/L,    0, -6*E*y^2/L^2,    2*E*y^2/L]
[  -E/L,            0,            0,  E/L,            0,            0]
[     0,-12*E*y^2/L^3, -6*E*y^2/L^2,    0, 12*E*y^2/L^3, -6*E*y^2/L^2]
[     0,  6*E*y^2/L^2,    2*E*y^2/L,    0, -6*E*y^2/L^2,    4*E*y^2/L]
```

这里要注意，由于前面的积分只对变量 x 进行积分，因此刚度矩阵中还包含变量 y。我们只要把 Ke 中第 2,3,5 和 6 行中的 y^2 换成截面惯性矩 I,并对第 1 和第 4 行乘截面积 A,即得到了如式(3-35)所示的单元刚度矩阵。

3.3 空间梁单元

若杆件系统的截面主轴或作用荷载不在同一平面内，则属于空间杆件系统问题。在一般情况下，梁单元每个结点的位移具有 6 个自由度，因而对应于 6 个结点力。在系统中取出结点为 i 和 j 的梁单元，如图 3-3 所示。同前，取右手坐标系，x 轴为单元轴线方向，而 y 轴和 z 轴为截面的主惯性轴。

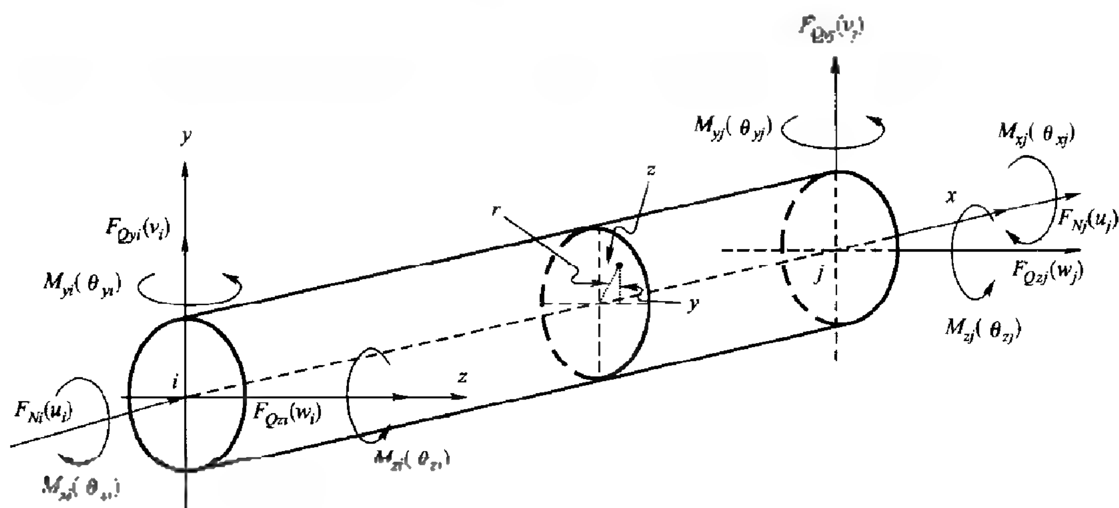


图 3-3 空间梁单元

将单元结点位移列阵记为

$$\boldsymbol{\delta}_i = \begin{bmatrix} u_i & v_i & w_i & \theta_{xi} & \theta_{yi} & \theta_{zi} \end{bmatrix}^{\mathrm{T}} \quad \boldsymbol{\delta}_j = \begin{bmatrix} u_j & v_j & w_j & \theta_{xj} & \theta_{yj} & \theta_{zj} \end{bmatrix}^{\mathrm{T}} \tag{3-36}$$

以及记

$$\boldsymbol{\delta}^e = \begin{bmatrix} \boldsymbol{\delta}_i^{\mathrm{T}} & \boldsymbol{\delta}_j^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{3-37}$$

相应的结点力列阵记为

$$\begin{aligned} \boldsymbol{f}_i &= \begin{bmatrix} F_{Ni} & F_{Qyi} & F_{Qzi} & M_{xi} & M_{yi} & M_{zi} \end{bmatrix}^{\mathrm{T}} \\ \boldsymbol{f}_i &= \begin{bmatrix} F_{Nj} & F_{Qyj} & F_{Qzj} & M_{xj} & M_{yj} & M_{zj} \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{3-38}$$

和

$$f = \begin{bmatrix} f_i^T & f_j^T \end{bmatrix}^T \tag{3-39}$$

由图 3-3 可以看出，F_{Ni} 和 F_{Nj} 表示作用于结点 i 和 j 的轴向力；F_{Qyi}、F_{Qzi}、F_{Qyj}、F_{Qzj} 表示 y 和 z 方向的剪力；M_{xi}、M_{xj} 表示扭矩；M_{yi}、M_{zi}、M_{yj}、M_{zj} 表示绕 y 和 z 轴的弯矩。与这些结点力相对应的位移，均在图中示出。图中所示的结点力和位移的方向均取为正方向。

通过与推导平面梁单元刚度矩阵类似的方法，可以得到空间梁单元的单元刚度矩阵。轴向位移 u 和扭转角 θ_x 的位移模式取 x 的线性函数，而挠度 v 和 w 则用三次多项式来表示。于是有

$$\begin{aligned} u &= a_0 + a_1 x, \quad v = b_0 + b_1 x + b_2 x^2 + b_3 x^3 \\ \theta_x &= e_0 + e_1 x, \quad w = c_0 + c_1 x + c_2 x^2 + c_3 x^3 \end{aligned} \tag{3-40}$$

另外可以将轴向结点位移、结点挠度和转角记为

$$\begin{aligned} \boldsymbol{\delta}_u &= \begin{bmatrix} u_i & u_j \end{bmatrix}^T, \quad \boldsymbol{\delta}_v = \begin{bmatrix} v_i & \theta_{zi} & v_j & \theta_{zj} \end{bmatrix}^T \\ \boldsymbol{\delta}_\theta &= \begin{bmatrix} \theta_{xi} & \theta_{xj} \end{bmatrix}^T, \quad \boldsymbol{\delta}_w = \begin{bmatrix} w_i & \theta_{yi} & w_j & \theta_{yj} \end{bmatrix}^T \end{aligned} \tag{3-41}$$

经过与平面梁单元类似的推导过程，得到用结点位移表示的位移模式。它们的矩阵形式是

$$u = \boldsymbol{N}_u \boldsymbol{\delta}_u, \quad \theta_x = \boldsymbol{N}_\theta \boldsymbol{\delta}_\theta, \quad v = \boldsymbol{N}_v \boldsymbol{\delta}_v, \quad w = \boldsymbol{N}_w \boldsymbol{\delta}_w \tag{3-42}$$

式中

$$\boldsymbol{N}_u = \boldsymbol{N}_\theta = \boldsymbol{H}_1 \boldsymbol{A}_1^{-1}, \quad \boldsymbol{N}_v = \boldsymbol{N}_w = \boldsymbol{H}_2 \boldsymbol{A}_2^{-1} \tag{3-43}$$

这便是位移的形函数矩阵。

式(3-42)中单元内任意点的位移可由式(3-37)中的结点位移表示，即

$$\boldsymbol{u} = \begin{Bmatrix} u \\ v \\ w \\ \theta_x \end{Bmatrix} = \begin{bmatrix} \boldsymbol{H}_u \\ \boldsymbol{H}_v \\ \boldsymbol{H}_w \\ \boldsymbol{H}_\theta \end{bmatrix} \boldsymbol{A}^{-1} \boldsymbol{\delta}^e = \boldsymbol{N}\boldsymbol{\delta}^e \tag{3-44}$$

式中

$$\boldsymbol{H}_u = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3-45}$$

$$\boldsymbol{H}_v = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & x & 0 & x^2 & 0 & 0 & 0 & x^3 \end{bmatrix} \tag{3-46}$$

$$\boldsymbol{H}_w = \begin{bmatrix} 0 & 0 & 1 & 0 & x & 0 & 0 & 0 & x^2 & 0 & x^3 & 0 \end{bmatrix} \tag{3-47}$$

$$\boldsymbol{H}_\theta = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & x & 0 & 0 \end{bmatrix} \tag{3-48}$$

而

$$
\boldsymbol{A}=\begin{bmatrix}
1&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0\\
1&0&0&0&0&0&l&0&0&0&0&0\\
0&1&0&0&0&l&0&l^2&0&0&0&l^3\\
0&0&1&0&l&0&0&0&l^2&0&l^3&0\\
0&0&0&1&0&0&0&0&0&l&0&0\\
0&0&0&0&1&0&0&0&2l&0&3l^2&0\\
0&0&0&0&0&1&0&2l&0&0&0&3l^2
\end{bmatrix} \tag{3-49}
$$

梁单元受到拉压、弯曲和扭转变形后，它的正应变可以分成三部分：拉压应变 ε_0、弯曲应变 ε_{by} 和 ε_{bz}，扭转产生的剪应变为 γ，于是有

$$
\boldsymbol{\varepsilon}=\begin{Bmatrix}\varepsilon_0\\ \varepsilon_{by}\\ \varepsilon_{bz}\\ \gamma\end{Bmatrix}=\begin{Bmatrix}u'\\ -yv''\\ -zw''\\ r\theta'_x\end{Bmatrix}=\begin{bmatrix}\boldsymbol{H}'_u\\ -y\boldsymbol{H}''_v\\ -z\boldsymbol{H}''_w\\ r\boldsymbol{H}'_\theta\end{bmatrix}\boldsymbol{A}^{-1}\boldsymbol{\delta}^e=\boldsymbol{B}\boldsymbol{\delta}^e \tag{3-50}
$$

式中 y 和 z 是横截面上点的坐标，r 是点到 x 轴的距离(图 3-3)。

由 Hook 定理，就得到用结点位移表示单元应力的表达式

$$
\boldsymbol{\sigma}=\begin{Bmatrix}\sigma_0\\ \sigma_{by}\\ \sigma_{bz}\\ \tau\end{Bmatrix}=\begin{bmatrix}E\boldsymbol{H}'_u\\ -Ey\boldsymbol{H}''_v\\ -Ez\boldsymbol{H}''_w\\ Gr\boldsymbol{H}'_\theta\end{bmatrix}\boldsymbol{A}^{-1}\boldsymbol{\delta}^e=\boldsymbol{D}\boldsymbol{B}\boldsymbol{\delta}^e=\boldsymbol{S}\boldsymbol{\delta}^e \tag{3-51}
$$

式中

$$
\boldsymbol{D}=\mathrm{diag}(E,E,E,G)
$$

把应变矩阵 $\boldsymbol{B}$ 和弹性矩阵 $\boldsymbol{D}$ 代入单元刚度矩阵的一般公式(2-70)，经过积分运算，可以得到其显式为：

$$
\boldsymbol{K}^e=\iint \mathrm{d}A\int_0^l \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}\mathrm{d}x=
$$

$$
\begin{bmatrix}
\frac{EA}{l} & & & & & & & & & & & \\
0 & \frac{12EI_z}{l^3} & & & & & & & & & & \\
0 & 0 & \frac{12EI_y}{l^3} & & & & & & & & & \\
0 & 0 & 0 & \frac{GJ_k}{l} & & & & & & & & \\
0 & 0 & \frac{6EI_y}{l^2} & 0 & \frac{4EI_y}{l} & & & & & & & \\
0 & \frac{6EI_z}{l^2} & 0 & 0 & 0 & \frac{4EI_z}{l} & & & \text{对} & & \text{称} & \\
-\frac{EA}{l} & 0 & 0 & 0 & 0 & 0 & \frac{EA}{l} & & & & & \\
0 & -\frac{12EI_z}{l^3} & 0 & 0 & 0 & -\frac{6EI_z}{l^2} & 0 & \frac{12EI_z}{l^3} & & & & \\
0 & 0 & -\frac{12EI_y}{l^3} & 0 & -\frac{6EI_y}{l^2} & 0 & 0 & 0 & \frac{12EI_y}{l^3} & & & \\
0 & 0 & 0 & -\frac{GJ_k}{l} & 0 & 0 & 0 & 0 & 0 & \frac{GJ_k}{l} & & \\
0 & 0 & \frac{6EI_y}{l^2} & 0 & \frac{2EI_y}{l} & 0 & 0 & 0 & -\frac{6EI_y}{l^2} & 0 & \frac{4EI_y}{l} & \\
0 & \frac{6EI_z}{l^2} & 0 & 0 & 0 & \frac{2EI_z}{l} & 0 & -\frac{6EI_z}{l^2} & 0 & 0 & 0 & \frac{4EI_z}{l}
\end{bmatrix}
\tag{3-53}
$$

式中，$I_y = \iint z^2 \mathrm{d}A$ 和 $I_z = \iint y^2 \mathrm{d}A$ 是梁单元截面对 y 和 z 轴的主惯性矩，$J_k = \iint r^2 \mathrm{d}A$ 是横截面对 x 轴的极惯性矩。

下面是推导三维梁单元刚度矩阵的 MATLAB 程序段：

```
[illegible]
[illegible]
  Hv=[0 1 0 0 0 x 0 x^2  0  0  0  x^3 ];   [illegible]
  Hw=[0 0 1 0 x 0 0  0  x^2 [illegible]];
  Hq=[0 0 0 1 0 0 0  0  0  x  0  0 ];
```

```
A=[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
   0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
   0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0;
   0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0;
   0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0;
   0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0;
   1, 0, 0, 0, 0, 0, L, 0, 0, 0, 0, 0;
   0, 1, 0, 0, 0, L, 0, L^2, 0, 0, 0, L^3;
   0, 0, 1, [illegible], 0, 0, [illegible], L^3, 0, [illegible]
   0, 0, 0, 1, 0, 0, 0, [illegible]
   0, 0, 0, 0, 1, 0, 0, 0, 2*L, 0, 3*L^2, 0
   0, 0, 0, 0, 0, 1, 0, 2*L, 0, 0, 0, 3*L^2];
   H=[diff(Hu,x,1);
      -y*diff(Hv,x,2);
      -z*diff(Hw,x,2);
      r*diff(Hq,x,1)];
   B=H*inv(A);
   D=diag([E,E,E,G]);
   Ke=int(transpose(B)*D*B,x,0,L)
```

执行后得到单元刚度矩阵如下：

```
Ke=
   [ E/L, 0, 0, 0, 0, 0, -E/L, 0, 0, 0, 0, 0]
   [ 0, 12*E*y^2/L^3, 0, 0, 0, 6*E*y^2/L^2, 0, -12*E*y^2/L^
3, 0, 0, 0, 6*E*y^2/L^2]
   [ 0, 0, 12*z^2/L^3*E, 0, 6*z^2/L^2*E, 0, 0, 0, -12*z^2/L
^3*E, 0, 6*z^2/L^2*E, 0]
   [ 0, 0, 0, r^2/L*G, 0, 0, 0, 0, 0, -r^2/L*G, 0, 0]
   [ 0, 0, 6*z^2/L^2*E, 0, 4*z^2/L*E, 0, 0, 0, -6*z^2/L^
2*E, 0, 2*z^2/L*E, 0]
   [ 0, 6*E*y^2/L^2, 0, 0, 0, 4*E*y^2/L, 0, -6*E*y^2/L^2,
 0, 0, 0, 2*E*y^2/L]
   [ -E/L, 0, 0, 0, 0, 0, E/L, 0, 0, 0, 0, 0]
   [ 0, -12*E*y^2/L^3, 0, 0, 0, -6*E*y^2/L^2, 0, 12*E*y^
2/L^3, 0, 0, 0, -6*E*y^2/L^2]
```

```
[ 0, 0, -12*z^2/L^3*E, 0, -6*z^2/L^2*E, 0, 0, 0, 12*z^2/L^3*E, 0, -6*z^2/L^2*E, 0]
 [ 0, 0, 0, -r^2/L*G, 0, 0, 0, 0, 0, r^2/L*G, 0, 0]
 [ 0, 0, 6*z^2/L^2*E, 0, -2*z^2/L*E, 0, 0, 0, -6*z^2/L^2*E, 0, 4*z^2/L*E, 0]
 [ 0, 6*E*y^2/L^2, 0, 0, 0, 2*E*y^2/L, 0, -6*E*y^2/L^2, 0, 0, 0, 4*E*y^2/L]
```

然后把 y^2 z^2 和 r^2 分别替换成 Iz, Iy, Jk,并在第 1 和第 6 行乘面积 A,即得到如式(3-53)所示的三维梁单元的单元刚度矩阵。

3.4 等效结点力

前面曾经指出,式(2-71)中的列阵 $\boldsymbol{f}$ 是由作用在单元上的体积力和面力的等效结点力组合而成。所谓等效结点力,是指原非结点荷载按照虚功相等的原则分配到单元结点上的力。式(2-72)是它的一般公式,对于作用在杆件单元上的分布力,可以简写为

$$\boldsymbol{f}_s = \int_l \boldsymbol{N}^{\mathrm{T}} \boldsymbol{p}_s \mathrm{d}x \tag{3-54}$$

式中,$\boldsymbol{p}_s$ 是分布荷载列阵,$\boldsymbol{N}$ 是与 $\boldsymbol{p}_s$ 对应的位移形函数矩阵。

现在分别对分布轴向力、分布扭矩、分布横向力和分布弯曲力矩的等效结点力进行讨论如下。

1. 分布轴向力

分布轴向力 $p(x)$ 如图 3-4 所示,由式(3-23)知,分布轴向力所对应的轴向位移的形函数矩阵是 $\boldsymbol{N}_u$,因此按式(3-54),它的等效结点力为

$$\boldsymbol{f}_s = \int_0^l p(x) \boldsymbol{N}_u^{\mathrm{T}} \mathrm{d}x \tag{3-55}$$

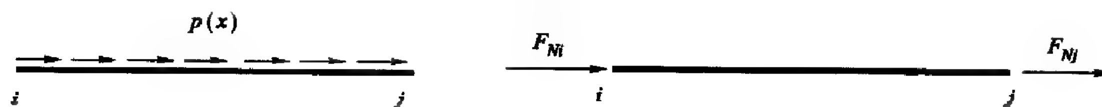


图 3-4 分布轴向力 $p(x)$ 的等效结点力

将式(3-24)代入上式,并把它详细写出,则得到

$$f_e = \begin{Bmatrix} F_{Ni} \\ F_{Nj} \end{Bmatrix} = \begin{bmatrix} 1 & -1/l \\ 0 & 1/l \end{bmatrix} \begin{Bmatrix} F_{p0} \\ F_{p1} \end{Bmatrix} \tag{3-56}$$

式中

$$F_{p0} = \int_0^l p(x)\mathrm{d}x, \quad F_{p1} = \int_0^l p(x)x\mathrm{d}x \tag{3-57}$$

在式(3-56)中，F_{Ni} 和 F_{Nj} 是等效结点轴向力。若 $p(x) = p = \text{const}$，即为均布轴向力，则 $F_{Ni} = F_{Nj} = pl/2$。可以看出，对于均布轴向力其等效结点力是总荷载的一半。对于线性分布的轴向力，并设在 i 结点的值为 p_1，在 j 结点的值为 p_2，那么经过简单的积分和矩阵乘法，可以得到等效结点力为

$$F_{Ni} = \frac{l}{6}(2p_1 + p_2), \quad F_{Nj} = \frac{l}{6}(p_1 + 2p_2) \tag{3-58}$$

推导线性分布轴力的等效结点力公式的 MATLAB 程序如下:

```
[illegible]
Nu=[1  0
   -1  1];
xx=[1 x];
[illegible]
x2=L;
L1=(x-x2)/(x1-x2);
L2=(x-x1)/(x2-x1);
p=p1*L1+p2*L2;
Fe=int(transpose(Nu)*p,x,0,L);
Fe=simple(Fe);
```

MATLAB 给出的等效结点力公式是:

```
Fe=
   [1/6*L*(p2+2*p1)]
   [1/6*L*(2*p2+p1)]
```

即为式(3-58)所示的等效结点力公式。如果推导其他分布形式的等效结点力，只要修改前面程序中的 p 的形式。比如要推导二次抛物线分布轴力的等效结点力，只要把 p 写成二次的 Lagrange 插值形式，即:

```
syms L x x1 x2 x3 p1 p2 p3
Nu=[1  0
```

```
H1 = [1 x] ;
Nu = H1* inv(Au) ;
x1 = 0 ;
x2 = L/2 ;
x3 = L ;
L1 = (x - x2)* (x - x3)/(x1 - x2)/(x1 - x3) ;
L2 = (x - x1)* (x - x3)/(x2 - x1)/(x2 - x3) ;
L3 = (x - x1)* (x - x2)/(x3 - x1)/(x3 - x2) ;
p = p1* L1 + p2* L2 + p3* L3 ;
fe = int(transpose(Nu)* p, x, 0, L);
fs = simple(fe) ;
```

执行后得到的等效结点力公式是:

```
fe =
[1/6* L* (p1 + 2* p2)]
[1/6* L* (2* p2 + p3)]
```

按照这个思路可以推导出更高阶次分布轴力的等效结点力。

2.分布扭矩

分布扭矩 $m_x(x)$ 的等效结点力如图 3-5 所示。因为扭转角的形函数矩阵和轴向位移的形函数矩阵是相同的，因此等效结点力的计算公式和轴向力的等效结点力相类同,即



图 3-5　分布扭矩 $m_x(x)$ 的等效结点力

$$f_t^e = \int_0^l m_x(x) N_u^{\mathrm{T}} \mathrm{d}x \tag{3-59}$$

把它详细写出,则得到

$$f_t^e = \begin{Bmatrix} M_{xi} \\ M_{xj} \end{Bmatrix} = \begin{bmatrix} 1 & -1/l \\ 0 & 1/l \end{bmatrix} \begin{Bmatrix} F_{m0} \\ F_{m1} \end{Bmatrix} \tag{3-60}$$

式中

$$F_{m0} = \int_0^l m_x(x)\mathrm{d}x, F_{m1} = \int_0^l m_x(x) x \mathrm{d}x \tag{3-61}$$

在式(3-60)中，M_{xi} 和 M_{xj} 是等效结点扭转力矩。前面已经指出，扭转角的形

函数矩阵与轴向位移的完全相同，因此根据等效结点力公式，扭转力矩的等效结点力形式上与具有相同分布形式的分布轴力的等效结点力公式相同，即对于均布扭转力矩，$m_x(x) = m_x = \text{const}$，则 $M_{xi} = M_{xj} = m_x l/2$ 。同样对于线性分布扭矩的等效结点力为

$$M_{xi} = \frac{l}{6}(2m_{x1} + m_{x2}), \quad M_{xj} = \frac{l}{6}(m_{x1} + 2m_{x2}) \tag{3-62}$$

式中 m_{x1} 和 m_{x2} 分别是两个结点上分布扭矩的集度值。前面给出的用来推导分布轴力等效结点力公式的 MATLAB 程序同样适用于分布扭矩。

3. 分布横向力

分布横向力 $p(x)$ 的等效结点力如图 3-6 所示。由式(3-23)知，分布横向力所对应挠度的形函数矩阵是 N_r 。因此它的等效结点力为

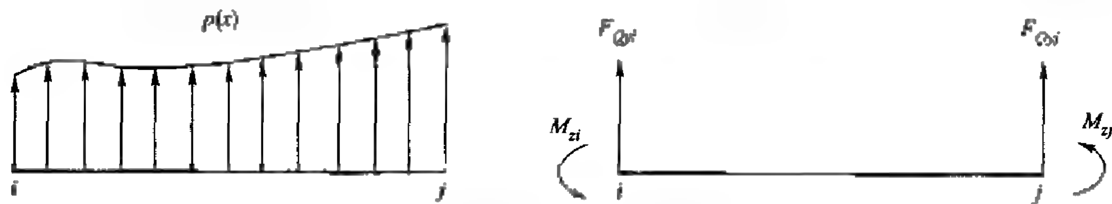


图 3-6 分布横向力 $p(x)$ 的等效结点力

$$f_s = \int_0^l p(x) N_r^{\mathrm{T}} \mathrm{d}x \tag{3-63}$$

将式(3-24)代入上式，并把它详细写出，则得到

$$f_s = \begin{Bmatrix} F_{Qyi} \\ M_{zi} \\ F_{Qyj} \\ M_{zj} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & -3/l^2 & 2/l^3 \\ 0 & 1 & -2/l & 1/l^2 \\ 0 & 0 & 3/l^2 & -2/l^3 \\ 0 & 0 & -1/l & 1/l^2 \end{bmatrix} \begin{Bmatrix} F_{p0} \\ F_{p1} \\ F_{p2} \\ F_{p3} \end{Bmatrix} \tag{3-64}$$

式中

$$\begin{aligned} F_{p0} &= \int_0^l p(x)\mathrm{d}x \qquad & F_{p1} &= \int_0^l p(x) x \mathrm{d}x \\ F_{p2} &= \int_0^l p(x) x^2 \mathrm{d}x \qquad & F_{p3} &= \int_0^l p(x) x^3 \mathrm{d}x \end{aligned} \tag{3-65}$$

而 F_{Qyi} 和 F_{Qyj} 是等效结点剪力，M_{zi} 和 M_{zj} 是等效结点弯矩。若为均布横向力，即 $p(x) = p$ ，于是

$$[F_{Qyi} \quad M_{zi} \quad F_{Qyj} \quad M_{zj}]^{\mathrm{T}} = \left[\frac{pl}{2} \quad \frac{pl^2}{12} \quad \frac{pl}{2} \quad -\frac{pl^2}{12}\right]^{\mathrm{T}} \tag{3-66}$$

对于线性分布的横向力，其等效结点力是

$$\begin{Bmatrix} F_{Qyi} \\ M_{zi} \\ F_{Qyj} \\ M_{zj} \end{Bmatrix} = \begin{Bmatrix} (7p_1 + 3p_2)l/20 \\ (3p_1 + 2p_2)l^2/60 \\ (3p_1 + 7p_2)l/20 \\ -(2p_1 + 3p_2)l^2/60 \end{Bmatrix} \tag{3-67}$$

式中 p_1 和 p_2 分别是两个结点处分布横向力的大小。

推导线性分布横向力等效结点力的 **MATLAB** 程序为:

```
syms L x x1 x2 p1 p2
Av=[ 1   0   0   0
     0   1   0   0
     1   L   L^2  L^3
     0   1   [illegible]];
H2=[1 x x^2 x^3];
Nv=H2*inv(Av);
x1=0;
x2=L;
l1=(x-x2)/(x1-x2);
l2=(x-x1)/(x2-x1);
p=p1*l1+p2*l2;
fe=int(transpose(Nv)*p,x,0,L);
fe=simple(fe)
```

执行后，得到等效结点力为:

```
fe =
[   1/20*L*(7*p1+3*p2)]
[ 1/60*L^2*(3*p1+2*p2)]
[   1/20*L*(3*p1+7*p2)]
[ -1/60*L^2*(2*p1+3*p2)]
```

很容易发现,它与式(3-67)一致。其他分布形式横向力的等效结点力公式可以参照前面分布轴向力同样的思路编程实现。

4. 分布弯曲力矩

分布弯曲力矩 $m_z(x)$ 等效结点力如图 3-7 所示。

同样道理，该荷载所对应的弯曲转角 v' 的形函数矩阵是 N'_v，它是 N_v 对 x 的导数，因此它的等效结点力为

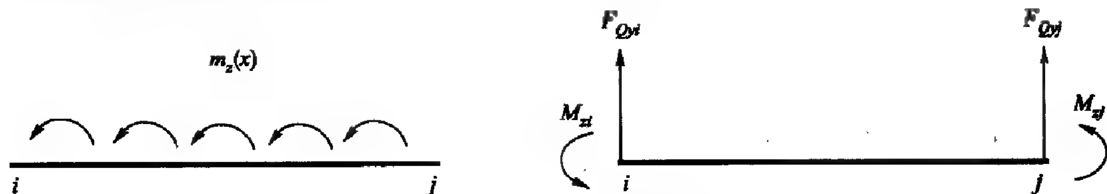


图 3-7　分布弯矩 $m_z(x)$ 等效结点力

$$f_s^e = \int_0^l m_z(x) N'^{\mathrm{T}}_v \mathrm{d}x \tag{3-68}$$

若把它详细写出，则和式(3-64)相同，但是式中

$$F_{p0} = 0 \qquad F_{p1} = \int_0^l m_z(x)\mathrm{d}x$$
$$F_{p2} = \int_0^l 2m_z(x)x\mathrm{d}x \qquad F_{p3} = \int_0^l 3m_z(x)x^2\mathrm{d}x \tag{3-69}$$

若为均布弯矩，即 $m_z(x) = m_z$，则

$$\begin{bmatrix} F_{Qyi} & M_{zi} & F_{Qyj} & M_{zj} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} -m_z & 0 & m_z & 0 \end{bmatrix}^{\mathrm{T}} \tag{3-70}$$

若为线性分布的弯矩，则为

$$\begin{Bmatrix} F_{Qyi} \\ M_{zi} \\ F_{Qyj} \\ M_{zj} \end{Bmatrix} = \begin{Bmatrix} -(m_{z1}+m_{z2})/2 \\ (m_{z1}-m_{z2})l/12 \\ (m_{z1}+m_{z2})/2 \\ -(m_{z1}-m_{z2})l/12 \end{Bmatrix} \tag{3-71}$$

式中 m_{z1} 和 m_{z2} 分别是两个结点上的分布弯矩的值。

分布弯矩等效结点力的推导与分布横向力类似，只要把挠度的形函数换成转角的形函数即可。线性分布弯矩等效结点力公式推导程序如下：

```
syms L x [illegible]
Av=[ 1 [illegible]
     0   1   0    0   [illegible]
     1   L   L^2  L^3   [illegible]
     0   1   2*L  3*L^2];
```

```
H2 = [1 x x^2 x^3] ;
Nv = H2* inv(Av) ;
x1 = 0 ;
x2 = L ;
L1 = (x - x2)/(x1 - x2) ;
L2 = (x - x1)/(x2 - x1) ;
m = m1* L1 + m2* L2 ;
fe = int(transpose(diff(Nv,x,1))* m, x, 0, L) ;
fe = simple(fe) ;
```

执行后，得到线性分布弯矩的等效结点力公式是:

```
fe =
[ -1/2* m1 -1/2* m2]
[ 1/12* (m1 - m2)* L]
[ 1/2* m1 + 1/2* m2]
[ -1/12* (m1 - m2)* L]
```

上面所示等效结点力即为式(3-71)。其他分布形式弯矩的等效结点力可参照分布轴力进行。

前面给出的是利用 MATLAB 的符号运算功能来推导等效结点力的公式的程序。但是在有限元程序中，应该利用 MATLAB 的数值运算功能,因为这些符号运算的速度很慢，会大大降低程序的计算效率。下面就给出平面梁单元线性分布轴力、横向力和弯矩的等效结点力计算函数。

根据前面推导的线性分布轴力、横向力和弯矩的等效结点力公式，可以很容易地用 MATLAB 编写实现这些功能的函数。

```
1.  function enf = EquivalentNodeForce( ie, p1, p2, idof )
2.  %   计算线性分布荷载的等效结点力
3.  %   输入参数:
4.  %       ie ------  单元号
5.  %       p1 ------  第一个结点上的分布力集度值
6.  %       p2 ------  第二个结点上的分布力集度值
7.  %       idof ----  分布力的种类,它可以是下面几种:
8.  %           1 ---  分布轴向力
9.  %           2 ---  分布横向力
10. %           3 ---  分布弯矩
```

```
11.  %   返回值：
12.  %       enf -----整体坐标系下等效结点力向量
13.    global gElement gNode
14.    enf = zeros( 6, 1 ) ;
15.    xi = gNode( gElement( ie, 1 ), 1 ) ;
16.    yi = gNode( gElement( ie, 1 ), 2 ) ;
17.    xj = gNode( gElement( ie, 2 ), 1 ) ;
18.    yj = gNode( gElement( ie, 2 ), 2 ) ;
19.    L = sqrt( (xj - xi)^2 + (yj - yi)^2 ) ;    计算单元长度
20.    switch idof
21.    case    1   % 分布轴向力
22.      enf( 1 ) = (2* p1 + p2)* L/6 ;
23.      enf( 4 ) = (p1 + 2* p2)* L/6 ;
24.    case    2   % 分布横向力
25.      enf( 2 ) = (7* p1 + 3* p2)* L/20 ;
26.      enf( 3 ) = (3* p1 + 2* p2)* L^2/60 ;
27.      enf( 5 ) = (3* p1 + 7* p2)* L/20 ;
28.      enf( 6 ) = - (2* p1 + 3* p2)* L^2/60 ;
29.    case    3   %   分布弯矩
30.      enf( 2 ) = - (p1 + p2)/2 ;
31.      enf( 3 ) = (p1 - p2)* L/12 ;
32.      enf( 5 ) = (p1 + p2)/2 ;
33.      enf( 6 ) = - (p1 - p2)* L/12 ;
34.    otherwise
35.      disp( sprintf('分布力的种类错误,单元号:% d',ie ) ) ;
36.    end
37.
38.    T = TransformMatrix( ie ) ;   %   计算单元的转换矩阵
39.       enf = T *  enf ;           %   把等效结点力转换到整体坐标下
40.return
```

第1句定义一个名为 EquivalentNodeForce 的函数，输入的参数是 ie,p1,p2,idof，返回值是 enf，它们的含义见程序的注释部分。MATLAB 程序中，% 是注释符号，类似于 C++ 语言中的//，表示该行%后面部分是注释。上面 2～12 行就是注释行，是函数的参数和返回值说明。第 13 行定义了两个全局变量 gElement 和 gNode。gElement 是一个矩阵，每一行表示一个单元，第 1 列是单元的第 1 个结点号，第 2 列是单元的第 2 个结点号。gNode 同样是一个矩阵，每一行表示一个结点，第 1 列是结点的 x 坐标，第 2 列是结点

向 y 坐标，第 15～19 行用来计算单元的长度，第 20～36 行根据载荷的类型，应用对应的等效结点力公式(3-62)、(3-67)或(3-71)来计算等效结点力。最后第 38 和 39 行是进行坐标转换的，可参考后面 3.5 节的内容。

3.5 单元刚度矩阵的坐标变换

在前面几节中，推导单元刚度矩阵时采用的全部是局部坐标系，它的坐标轴方向是由单元即杆或梁的截面主方向确定的。采用这样的坐标系，可以得到具有统一形式的单元刚度矩阵。但是，实际的结构一般由具有不同方向和处于不同位置的杆或梁所构成的。对于由不同方向单元组成的结构，它的整体刚度矩阵并不能由局部坐标下的单元刚度矩阵简单叠加而成，所以必须建立一个统一的整体坐标系。计算时先将单元上的结点力和位移转换到整体坐标系，单元刚度矩阵亦作坐标变换，之后可按叠加规则直接相加组成整体刚度矩阵。下面就来讨论坐标系的转换方法。

设 $\boldsymbol{f}'^e$、$\boldsymbol{\delta}'^e$ 和 $\boldsymbol{K}'^e$ 分别表示单元在局部坐标系 $O'x'y'z'$ 的结点力（包括等效结点力）、结点位移和刚度矩阵；$\boldsymbol{f}^e$、$\boldsymbol{\delta}^e$ 和 $\boldsymbol{K}^e$ 表示在整体坐标系 $Oxyz$ 中的结点力（包括等效结点力）、结点位移和刚度矩阵。于是有

$$\boldsymbol{f}'^e = \boldsymbol{K}'^e\boldsymbol{\delta}'^e \qquad \boldsymbol{f}^e = \boldsymbol{K}^e\boldsymbol{\delta}^e \tag{3-72}$$

如果以 $\boldsymbol{T}$ 表示结点力和结点位移在局部坐标系与整体坐标系之间的转换矩阵，则

$$\boldsymbol{f}^e = \boldsymbol{T}\boldsymbol{f}'^e, \quad \boldsymbol{\delta}^e = \boldsymbol{T}\boldsymbol{\delta}'^e \tag{3-73}$$

将式(3-72)的第一式代入式(3-73)的第一式，得

$$\boldsymbol{f}^e = \boldsymbol{T}\boldsymbol{K}'^e\boldsymbol{\delta}'^e \tag{3-74}$$

而把式(3-73)的第二式代入式(3-72)的第二式，又有

$$\boldsymbol{f}^e = \boldsymbol{K}^e\boldsymbol{T}\boldsymbol{\delta}'^e \tag{3-75}$$

于是

$$\boldsymbol{T}\boldsymbol{K}'^e\boldsymbol{\delta}'^e = \boldsymbol{K}^e\boldsymbol{T}\boldsymbol{\delta}'^e \tag{3-76}$$

由于 $\boldsymbol{\delta}'^e$ 的任意性，上式可写成

$$\boldsymbol{K}^e\boldsymbol{T} = \boldsymbol{T}\boldsymbol{K}'^e \tag{3-77}$$

上式两边各右乘 $\boldsymbol{T}^{\mathrm{T}}$ 便可得到

$$\boldsymbol{K}^e\boldsymbol{T}\boldsymbol{T}^{\mathrm{T}} = \boldsymbol{T}\boldsymbol{K}'^e\boldsymbol{T}^{\mathrm{T}} \tag{3-78}$$

由于转换矩阵是正交矩阵，即 $TT^{T} = I$ 所以便可求得整体坐标系下的单元刚度矩阵为

$$K^{e} = TK'^{e}T^{T} \tag{3-79}$$

因此，只要得到转换矩阵 T，就可以在局部坐标系和整体坐标系之间进行结点力，结点位移和单元刚度矩阵的转换。

3.5.1 平面杆单元的坐标转换矩阵

对于平面杆单元，取整体坐标轴 $Oxyz$，将 x 和 y 轴置于杆系平面内，z 轴垂直于杆系平面。整体坐标轴与局部坐标轴 $x'y'$ 相交成 α 角，如图 3-8 所示。

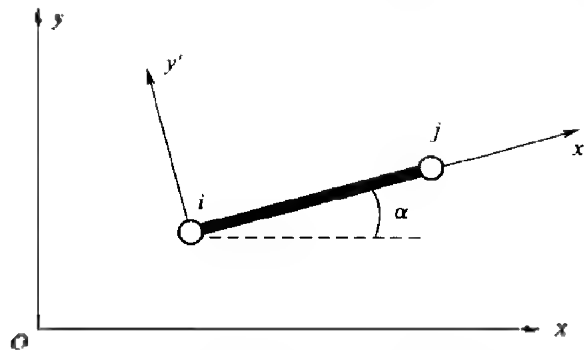


图 3-8 整体和局部坐标轴的位置

考虑单元结点 i，在局部坐标系下其位移为 u'_i，它与整体坐标中相应的结点位移 u_i 和 v_i 之间存在着几何关系

$$u_i = u'_i\cos\alpha, \quad v_i = u'_i\sin\alpha \tag{3-80}$$

对于结点 j，具有同样的几何关系，因此可写成如下的矩阵形式

$$\begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix} = \begin{bmatrix} \cos\alpha & 0 \\ \sin\alpha & 0 \\ 0 & \cos\alpha \\ 0 & \sin\alpha \end{bmatrix} \begin{Bmatrix} u'_i \\ u'_j \end{Bmatrix} \tag{3-81}$$

因此转换矩阵为

$$T = \begin{bmatrix} \cos\alpha & 0 \\ \sin\alpha & 0 \\ 0 & \cos\alpha \\ 0 & \sin\alpha \end{bmatrix} \tag{3-82}$$

如果这两个结点在整体坐标系下的坐标分别是 (x_i, y_i) 和 (x_j, y_j)，那么有

$$\cos\alpha = \frac{x_j - x_i}{l}, \quad \sin\alpha = \frac{y_j - y_i}{l} \tag{3-83}$$

而单元的长度 l 则为

$$l = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

这样就通过结点在整体坐标系下的坐标，得到了平面杆单元的转换矩阵。

3.5.2 空间杆单元的转换矩阵

如图 3-9 所示，设有空间杆单元，取单元轴线为局部坐标系的 x' 轴，并设它在整体坐标系中的方向余弦为 (l_x, m_x, n_x) 。

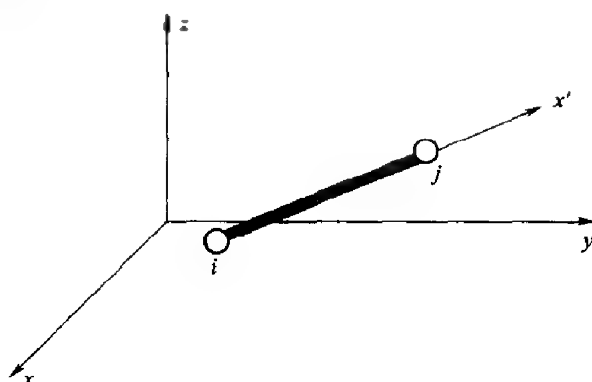


图 3-9 空间杆单元

在局部坐标系下的位移 u'_i 与整体坐标中相应的结点位移 u_i 、v_i 和 w_i 之间存在着几何关系

$$u_i = u'_i l_x, \quad v_i = u'_i m_x, \quad w_i = u'_i n_x \tag{3-85}$$

对于结点 j ，具有同样的几何关系，因此可写成如下的矩阵形式

$$\begin{Bmatrix} u_i \\ v_i \\ w_i \\ u_j \\ v_j \\ w_j \end{Bmatrix} = \begin{bmatrix} l_x & 0 \\ m_x & 0 \\ n_x & 0 \\ 0 & l_x \\ 0 & m_x \\ 0 & n_x \end{bmatrix} \begin{Bmatrix} u'_i \\ u'_j \end{Bmatrix} \tag{3-86}$$

因此转换矩阵为

$$\boldsymbol{T} = \begin{bmatrix} l_x & m_x & n_x & 0 & 0 & 0 \\ 0 & 0 & 0 & l_x & m_x & n_x \end{bmatrix}^{\mathrm{T}} \tag{3-87}$$

如果单元的结点在整体坐标系下的坐标分别是 (x_i, y_i, z_i) 和 (x_j, y_j, z_j)，那么有

$$l_x = \frac{x_j - x_i}{l}, \quad m_x = \frac{y_j - y_i}{l}, \quad n_x = \frac{z_j - z_i}{l} \tag{3-88}$$

而单元的长度 l 则为

$$l = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} \tag{3-89}$$

同样通过结点在整体坐标系下的坐标，得到了空间杆单元的转换矩阵。

3.5.3 平面梁单元的转换矩阵

对于平面梁单元，同样可以取如图 3-8 所示的局部坐标系和整体坐标系。梁单元的结点 i 在局部坐标系下的位移有三个，即 u'_i、v'_i 和 θ'_i，它们与整体坐标中相应的结点位移 u_i、v_i 和 θ_i 之间存在着几何关系

$$u_i = u'_i\cos\alpha - v'_i\sin\alpha$$

$$v_i = u'_i\sin\alpha + v'_i\cos\alpha$$

$$\theta_i = \theta'_i$$

如果写成矩阵形式，得

$$\begin{Bmatrix} u_i \\ v_i \\ \theta_i \end{Bmatrix} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u'_i \\ v'_i \\ \theta'_i \end{Bmatrix} \tag{3-91}$$

对于结点 j，有相同的关系式，因此平面梁单元的转换矩阵为

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_1 & 0 \\ 0 & \boldsymbol{T}_1 \end{bmatrix} \tag{3-92}$$

式中

$$\boldsymbol{T}_1 = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3-93}$$

式中的 $\sin\alpha$ 和 $\cos\alpha$ 同样由式(3-83)和式(3-84)确定。

下面给出计算平面梁单元坐标转换矩阵的 MATLAB 程序，因为坐标转换矩阵在单元刚度矩阵转换、等效结点力转换和计算结点内力时都要用到，因此可定义一个函数，以方便

调用。

```
1.  function T = TransformMatrix( ie )
2.  %   计算单元的坐标转换矩阵( 局部坐标 -> 整体坐标 )
3.  %   输入参数
4.  %      ie - - - - - -结点号                                  注释
5.  %   返回值
6.  %      T - - - - - - - -从局部坐标到整体坐标的坐标转换矩阵
7.      global gElement gNode
8.      xi = gNode( gElement( ie, 1 ), 1 ) ;
9.      yi = gNode( gElement( ie, 1 ), 2 ) ;
10.     xj = gNode( gElement( ie, 2 ), 1 ) ;
11.     yj = gNode( gElement( ie, 2 ), 2 ) ;
12.     L = sqrt( (xj - xi)^2 + (yj - yi)^2 ) ;
13.     c = (xj - xi)/L ;                                        计算方向余弦
14.     s = (yj - yi)/L ;
15.    T =[c -s  0  0   0     0
16.        c s   0  0   0     0
17.        0 0   1  0   0     0                                  构造变换矩阵
18.        0 0   0  c   -s    0
19.        0 0   0  s   c     0
20.        0 0   0  0   0     1];
21.  return
```

第1句定义一个名为TransformMatrix的函数,输入的参数是ie,返回值是T。其后2~6行全部是注释,用来说明该函数的主要功能以及输入输出参数的含义。第7行定义两个全局变量gElement和gNode,它们的含义与前面计算等效结点力的函数中的一样。根据式(3-88),只要得到梁单元两个结点的坐标,即可计算梁单元的坐标转换矩阵。第8~11行从gNode和gElement中得到两个结点的坐标。第12~14行得到了梁单元的长度及其倾斜角度的正弦和余弦。第15~20行生成了单元的坐标转换矩阵,并把它保存于变量T中,在执行函数返回语句return后,就作为返回值传递给了调用该函数的程序段。

3.5.4 空间梁单元的转换矩阵

实际上,对于向量在不同坐标系之间的转换关系有统一的公式。设局部坐标系的3个轴x'、y'和z'在整体坐标系$Oxyz$中的方向余弦分别是(l_x,m_x,n_x)、(l_y,m_y,n_y)和(l_z,m_z,n_z),那么一个在局部坐标系中定义的向量与它相

应的整体坐标系下的向量的转换矩阵为[3]

$$T_2 = \begin{bmatrix} l_x & l_y & l_z \\ m_x & m_y & m_z \\ n_x & n_y & n_z \end{bmatrix} \tag{3-94}$$

它的第1列、第2列和第3列分别是 x' 轴，y' 轴和 z' 轴在整体坐标系中的3个方向余弦。由于空间梁单元的每个结点都有6个位移，可组成两个三维的向量，因此它的结点位移共有4个三维的向量，转换矩阵相应地为

$$T = \begin{bmatrix} T_2 & 0 & 0 & 0 \\ 0 & T_2 & 0 & 0 \\ 0 & 0 & T_2 & 0 \\ 0 & 0 & 0 & T_2 \end{bmatrix} \tag{3-95}$$

当利用式(3-94)计算转换矩阵 T 时，必须已知局部坐标系 $O'x'y'z'$ 的三个轴在整体坐标系 $Oxyz$ 的方向余弦。一般说来，对于梁单元的轴线方向 x' 轴，它的方向余弦是能够通过两个结点的坐标直接求得的，但是对于截面主轴 y' 和 z' 的方向余弦，必须给出附加信息，才能确定。现在来推导梁单元转换矩阵 T 的转换公式。

1. x' 轴在 $Oxyz$ 坐标系中的方向余弦

设 (x_i, y_i, z_i) 和 (x_j, y_j, z_j) 为结点 i 和 j 在整体坐标系 $Oxyz$ 中的坐标，如图3-10所示。

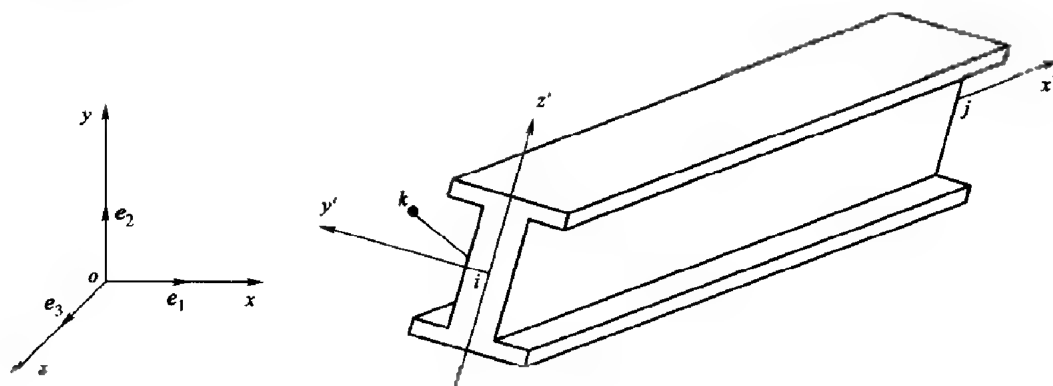


图3-10 局部坐标和整体坐标的变换

x' 轴的3个方向余弦跟空间杆单元的一样，见式(3-88)。若以 e_1、e_2 和 e_3 表示整体坐标系下沿3个坐标轴方向的单位矢量，则 x' 轴的单位矢量 e'_1 可以写成

$$e'_1 = l_x e_1 + m_x e_2 + n_x e_3 \tag{3-96}$$

2. z' 轴在 $Oxyz$ 坐标系中的方向余弦

在梁单元的主形心惯性平面（$x'y'$ 平面）上取一点 k，该点在整体坐标系中的坐标为 x_k、y_k 和 z_k。若用 $\boldsymbol{g}$ 表示 ik 方向的单位矢量，则有

$$\boldsymbol{g} = g_1 e_1 + g_2 e_2 + g_3 e_3 \tag{3-97}$$

式中

$$g_1 = \frac{x_k - x_i}{l_k}, \quad g_2 = \frac{y_k - y_i}{l_k}, \quad g_3 = \frac{z_k - z_i}{l_k} \tag{3-98}$$

而

$$l_k = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2} \tag{3-99}$$

根据矢量的外积法则，z' 轴的单位矢量 e'_3 可以写成

$$e'_3 = \frac{e'_1 \times \boldsymbol{g}}{|e'_1 \times \boldsymbol{g}|} = l_z e_1 + m_z e_2 + n_z e_3 \tag{3-100}$$

把式(3-96)和式(3-97)代入式(3-100)，得到

$$l_z = \frac{m_x g_3 - n_x g_2}{s}, \quad m_z = \frac{n_x g_1 - l_x g_3}{s}, \quad n_z = \frac{l_x g_2 - m_x g_1}{s} \tag{3-101}$$

式中

$$s = \sqrt{(m_x g_3 - n_x g_2)^2 + (n_x g_1 - l_x g_3)^2 + (l_x g_2 - m_x g_1)^2} \tag{3-102}$$

3. y' 轴在 $Oxyz$ 坐标系中的方向余弦

按照矢量乘法，y' 轴在 $Oxyz$ 坐标系中的方向余弦是

$$e'_2 = e'_3 \times e'_1 = l_y e_1 + m_y e_2 + n_y e_3 \tag{3-103}$$

把式(3-96)和式(3-100)代入式(3-102)，得到

$$l_y = m_z n_x - n_z m_x, \quad m_y = n_z l_x - l_z n_x, \quad n_y = l_z m_x - l_x m_z \tag{3-104}$$

综合上述计算，得到局部坐标系和整体坐标系之间位移的转换矩阵为

$$T_2 = \begin{bmatrix} l_x & \dfrac{g_1 - l_x(l_x g_1 + m_x g_2 + n_x g_3)}{s} & \dfrac{m_x g_3 - n_x g_2}{s} \\ m_x & \dfrac{g_2 - m_x(l_x g_1 + m_x g_2 + n_x g_3)}{s} & \dfrac{n_x g_1 - l_x g_3}{s} \\ n_x & \dfrac{g_3 - n_x(l_x g_1 + m_x g_2 + n_x g_3)}{s} & \dfrac{l_x g_2 - m_x g_1}{s} \end{bmatrix} \tag{3-105}$$

因此，在计算空间梁单元的转换矩阵时，必须提供附加点 k 的坐标，来确定梁单元的主平面。这是与平面梁单元和杆单元的不同之处，在程序设计时应予以考虑。

3.6 整体刚度矩阵

在得到整体坐标系中的单元刚度矩阵之后，可以按照叠加法（直接刚度法）组装成整体刚度矩阵[4,5,6]。为此，必须把有关矩阵作适当的扩大改写，使得所有单元的刚度矩阵具有统一的格式才可进行。

不失一般性，设整个系统被离散成 n_e 个平面梁单元和 n 个结点，整个杆件系统的结点位移列阵为

$$\boldsymbol{\delta}_{3n\times1} = [\boldsymbol{\delta}_1^{\mathrm{T}} \quad \boldsymbol{\delta}_2^{\mathrm{T}} \quad \cdots \quad \boldsymbol{\delta}_n^{\mathrm{T}}]^{\mathrm{T}} \tag{3-106}$$

上式是由各结点位移按照结点的编号从小到大排列组成的。再将原来的六阶方阵 $\boldsymbol{K}^e$ 加以扩大，写成 $3n\times3n$ 的方阵如下

$$\begin{array}{ccccccc} & & \cdots & i & \cdots & j & \cdots \\ \boldsymbol{K}^e_{3n\times3n} = & \left[\begin{array}{ccccc} & \vdots & & \vdots & \\ \cdots & \boldsymbol{K}_{ii} & \cdots & \boldsymbol{K}_{ij} & \cdots \\ & \vdots & & \vdots & \\ \cdots & \boldsymbol{K}_{ji} & \cdots & \boldsymbol{K}_{jj} & \cdots \\ & \vdots & & \vdots & \end{array}\right] & \begin{array}{c} \vdots \\ i \\ \vdots \\ j \\ \vdots \end{array} \end{array} \tag{3-107}$$

式中虚点和空处的元素为 3×3 的零矩阵，而矩阵记号外面的 i 和 j 表示分块意义下，子矩阵 $\boldsymbol{K}_{ii}$ 等所在的行和列（实际上是 3 行 3 列），例如 $\boldsymbol{K}_{ij}$，这个三阶方阵就放在分块意义下的第 i 行和第 j 列的位置上，这里 i 和 j 的顺序也是按照结点编号从小到大排列的。显然，经过这样扩大后的方阵（3-107）仍是对称的，而且所有其他单元刚度矩阵中的 4 个子矩阵都可按照其下标（由单元的结点的编号组成）在式 3-107 的格式中“对号叠加”。

同样，把单元的等效结点力加以扩大，改写成 $3n\times1$ 阶的列阵

$$\begin{array}{cc} & \quad\cdots \quad i \quad \cdots \quad j \quad \cdots \\ \boldsymbol{f}^e_{3n\times1} = & \left[\cdots \quad \boldsymbol{f}_i^e \quad \cdots \quad \boldsymbol{f}_j^e \quad \cdots\right]^{\mathrm{T}} \end{array} \tag{3-108}$$

之后，就可以按照叠加规则直接相加，得到整体刚度矩阵和结点力列阵，即

$$\boldsymbol{K} = \sum_{e=1}^{n_e} \boldsymbol{K}^e, \quad \boldsymbol{f} = \sum_{e=1}^{n_e} \boldsymbol{f}^e \tag{3-109}$$

最后得到整个结构的平衡方程

$$K\delta = f \tag{3-110}$$

式中 δ 是整个结构的结点位移列阵；f 是结点荷载列阵，它已经包含了非结点荷载的等效结点力；K 是结构的整体刚度矩阵。

式(3-110)中，K、f 和 δ 中各元素的排列顺序应注意按相同的顺序，一般是按照结点编号，从小到大排列。式(3-110)实际上是包含 $3n$ 个以结点位移分量为基本未知量的线性代数方程组，它们是在 n 个结点上列出的 $3n$ 个平衡方程。

下面给出计算单元刚度矩阵的 MATLAB 程序。

```
1.function k = StiffnessMatrix( ie, icoord )
2.  %   计算单元刚度矩阵
3.  %   输入参数:
4.  %     ie --------  单元号
5.  %     icoord ---  坐标系参数,可以是下面两个之一
6.  %                 1  ----  整体坐标系
7.  %                 2  ----  局部坐标系
8.  %   返回值:
9.  %     k  ----------  根据 icoord 的值,相应坐标系下的刚度矩阵
10.   global gNode gElement gMaterial
11.   k = zeros( 6, 6 ) ;
12.   E = gMaterial( gElement(ie, 3), 1 ) ;
13.   I = gMaterial( gElement(ie, 3), 2 ) ;
14.   A = gMaterial( gElement(ie, 3), 3 ) ;
15.   xi = gNode( gElement( ie, 1 ), 1 ) ;
16.   yi = gNode( gElement( ie, 1 ), 2 ) ;
17.   xj = gNode( gElement( ie, 2 ), 1 ) ;
18.   yj = gNode( gElement( ie, 2 ), 2 ) ;
19.   L = ( (xj - xi)^2 + (yj - yi)^2 )^(1/2) ;
20.   k = [E* A/L      0             0           -E* A/L     0             0
21.        0      12* E* I/L^3   6* E* I/L^2    0   -12* E* I/L^3   6* E* I/L^2
22.        0       6* E* I/L^2   4* E* I/L      0    -6* E* I/L^2   2* E* I/L
23.     -E* A/L      0             0            E* A/L     0             0
24.        0     -12* E* I/L^3  -6* E* I/L^2    0    12* E* I/L^3  -6* E* I/L^2
```

```
25.         0    6* E* I/L^2   2* E* I/L  0   -6* E* I/L^2  4* E* I/L] ;
26.     if icoord == 1
27.         T = TransformMatrix( ie ) ;
28.         k = T* k* transpose(T) ;
29.     end
30. [illegible]
```

第1行定义了计算单元刚度矩阵的函数 StiffnessMatrix，输入参数是 ie 和 icoord，返回值是 k。它们的含义见第2~9行的注释。第10行定义了3个全局变量 gNode、gElement 和 gMaterial。前面两个已经作过说明，gMaterial 是材料矩阵，每一行表示一种材料，它包括弹性模量 E、抗弯惯性矩 I 和截面积 A。第15~19行计算了单元的长度，第20~25行是根据公式(3-35)构造局部坐标系下单元刚度矩阵，第26~29行是根据输入参数 icoord 的值，对单元刚度矩阵进行坐标转换，并把它保存到返回值变量 k 中。

用 MATLAB 编程实现单元刚度矩阵的"对号叠加"的关键是根据结点号和自由度号来得到在整体刚度矩阵中的行号和列号。因为每一个结点都有3个自由度，所以结点号为 i 的第 p 个自由度在整体刚度矩阵中的行号（或列号）就是 m = (i - 1) * 3 + p，因此我们可以写出集成单元刚度矩阵到整体刚度矩阵的 MATLAB 函数为：

```
1.  function [illegible]( ie, k )
2.  %  把单元刚度矩阵集成到整体刚度矩阵
3.  %  输入参数:
4.  %      ie  ---  单元号
5.  %      k   ---  单元刚度矩阵
6.  %  返回值:
7.  %      无
8.      global gElement gK
9.      for i = 1:1:2
10.         for j = 1:1:2
11.             for p = 1:1:3
12.                 for q = 1:1:3
13.                     m = (i-1)*3+p ;
14.                     n = (j-1)*3+q ;
15.                     M = (gElement(ie,i) -1)* 3 + p ;
16.                     N = (gElement(ie,j) -1)* 3 + q ;
17.                     gK(M,N) = gK(M,N) + k(m,n) ;
18.                 end
19.             end
20.         end
```

```
21 [illegible] end [illegible]
22 [illegible]
```

该函数的核心就是第 13～16 行中的自由度转换公式。

整体刚度矩阵具有很多重要性质，下面逐一说明：

(1)整体刚度矩阵 $\boldsymbol{K}$ 中每一元素称为刚度系数，它的物理意义是要使结构体的某一结点自由度发生单位广义位移，而其他结点自由度都保持零位移的状态下，所有各结点需要施加的结点广义力。这可以从式(3-110)中令某一结点自由度为 1,而其余结点的所有位移分量为零得到。

(2)整体刚度矩阵的对角元上的主元素都是正的。例如，由性质(1)可知,刚度矩阵 $\boldsymbol{K}$ 中的元素 k_{44}表示结点 2 在 x 方向产生单位位移而其他位移均为零时,在结点 2 的 x 方向上施加的力，它自然应顺着位移的方向，因而为正号。

(3)整体刚度矩阵 $\boldsymbol{K}$ 是对称矩阵。这可以从单元刚度矩阵的定义式(2-70)看出。因此在实际计算中只需要计算对角元及其某一边的元素。

(4)整体刚度矩阵 $\boldsymbol{K}$ 是一个稀疏阵,如果遵守一定的结点编号规则,可使非零元素都集中于主对角线附近而呈带状。

(5)整体刚度矩阵 $\boldsymbol{K}$ 是一个奇异矩阵。因为物体在受到平衡力作用时，可以是静止不动,但也可以作匀速运动,即物体的绝对位移不能确定,也就是说整体刚度矩阵 $\boldsymbol{K}$ 不存在逆矩阵,因此它是奇异矩阵。在排除刚体位移后,它是正定矩阵。

3.7 边界约束条件的处理

在上一节讨论整体刚度矩阵的奇异性时已经指出，必须考虑边界约束条件,排除系统的刚体位移,才能从方程组(3-110)求解结点位移。一般情况下，所考虑问题的边界往往已有一定的位移约束条件，可以直接利用，有时也需要适当指定某些结点在某些方向的位移值，以避免出现刚体运动。我们可以把这类约束条件称为第一类约束条件,即

$$\delta_i = \beta_i \tag{3-111}$$

它是应用最广泛也是处理比较简单的一种。在理论上，引用这些约束条件后，待求结点未知量的数目和方程的阶数便可相应地减少。然而，在实际程序实现时，一般希望保持方程阶数不变，以避免程序过于复杂。为了说明引入

第一类约束条件的方法，不妨考察下面只有6个未知数的方程。方程(3-110)展开成如下的形式

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{Bmatrix} \tag{3-112}$$

设结点位移 δ_1 和 δ_4 被指定为

$$\delta_1 = \beta_1, \quad \delta_4 = \beta_4 \tag{3-113}$$

为了满足式(3-113)，可以将方程(3-112)改写成如下形式

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{22} & k_{23} & 0 & k_{25} & k_{26} \\ 0 & k_{32} & k_{33} & 0 & k_{35} & k_{36} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & k_{52} & k_{53} & 0 & k_{55} & k_{56} \\ 0 & k_{62} & k_{63} & 0 & k_{65} & k_{66} \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{Bmatrix} = \begin{Bmatrix} \beta_1 \\ f_2 - k_{21}\beta_1 - k_{24}\beta_4 \\ f_3 - k_{31}\beta_1 - k_{34}\beta_4 \\ \beta_4 \\ f_5 - k_{51}\beta_1 - k_{54}\beta_4 \\ f_6 - k_{61}\beta_1 - k_{64}\beta_4 \end{Bmatrix} \tag{3-114}$$

显然，用这组维数不变的方程来求解结点位移，其解为 $\delta_1 = \beta_1$ 和 $\delta_4 = \beta_4$，而其余位移分量仍为原来的解。这种方法称为"划行划列法"。

把上面这个划行划列的方法用 **MATLAB** 程序段来表示，则为

```
1.    [bc1_number,dummy] = size(gBC1);
2.    for ibc=1:1:bc1_number
3.        n = gBC1(ibc, 1);
4.        d = gBC1(ibc, 2);
5.        m = (n-1)*3 + d;
6.        f = f - gBC1(ibc,3) * gK(:,m);
7.        f(m) = gBC1(ibc,3);
8.        gK(:,m) = zeros(node_number*3,1);
9.        gK(m,:) = zeros(1, node_number*3);
```

```
10    gK(m,m) = 1.0;
11  end
```

这里 gK 是整体刚度矩阵，f 是右端整体结点力向量；node _ number 是一个标量，表示有限元模型的结点数。gBC1 是保存第一类边界条件的矩阵，每一行表示一个边界条件。它的第 1 列是结点号，第 2 列是自由度号，第 3 列是自由度的约束值，即 β 值。第 1 行是得到第一类边界条件的数目，从第 2 ~ 11 行对所有边界条件作循环，进行划行划列处理。其中第 5 行是把第 n 个结点的第 d 个自由度换算成整体刚度矩阵中的第 m 行，换算公式跟前面单元刚度矩阵集成整体刚度矩阵时的相同，即 m = (n - 1) * 3 + d。

另一种考虑指定结点位移的方法，是将 $\boldsymbol{K}$ 中与指定结点位移有关的主对角元乘上一个大数 λ，同时将 $\boldsymbol{f}$ 的对应元素换上结点位移指定值与同一个大数以及主对角元相乘的乘积。实际上，这种方法就是使得 $\boldsymbol{K}$ 中相应行的修正项远大于非修正项。

若用此方法来修正上面的例子，则方程(3-112)将成为

$$\begin{bmatrix} k_{11}\lambda & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} \\ k_{41} & k_{42} & k_{43} & k_{44}\lambda & k_{45} & k_{46} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{Bmatrix} = \begin{Bmatrix} \beta_1 k_{11} \lambda \\ f_2 \\ f_3 \\ \beta_2 k_{44} \lambda \\ f_5 \\ f_6 \end{Bmatrix} \tag{3-115}$$

为了看出此方程组能给出所需的结果，我们来考察它的第一个方程

$$k_{11}\lambda\delta_1 + k_{12}\delta_2 + k_{13}\delta_3 + k_{14}\delta_4 + k_{15}\delta_5 + k_{16}\delta_6 = \beta_1 k_{11} \lambda \tag{3-116}$$

从数值计算的结果来看，它与下式相同

$$k_{11}\lambda\delta_1 = k_{11}\lambda\beta_1 \tag{3-117}$$

这是因为

$$k_{11}\lambda\delta_1 \gg k_{1j}\delta_j \quad (j = 2,3,4,5,6) \tag{3-118}$$

这种方法称为“乘大数法”。这种方法也保持了原来刚度矩阵的稀疏、带状和对称等特性。为了更好地理解它，下面给出了实现“乘大数法”的 MATLAB 的程序段。可以看出在程序实现上“乘大数法”比“划行划列法”更简单。

乘大数法的 MATLAB 程序可以这样写：

```
1  [bc1_number,dummy] = size(gBC1);
2  for ibc=1:1:bc1_number
```

```
3.        n = gBC1(ibc, 1 ) ;
4.        d = gBC1(ibc, 2 ) ;
5.        m = (n-1)* 3 + d ;
6.        f(m) = gBC1(ibc, 3)* gK(m,m) * 1e15 ;
7.        gK(m,m) = gK(m,m) * 1e15 ;
8.    end
```

其中变量的意义跟前面划行划列法的程序중相同，我们发现从程序实现上来看，乘大数法要比划行划列法简单得多，但是大数的选择要根据经验，也可以采用考题试算的办法，得到合适的大数值。

第二类约束条件是指两个自由度之间有某种约束，不失一般性，可以把它写成[7]

$$\delta_i + \alpha\delta_j = \beta \tag{3-119}$$

这里，只要令 $\alpha = 0$ ，第二类约束条件即退化为第一类。这种约束条件可用来模拟两个结点位移之间的约束关系，这在桥梁结构分析中经常碰到，例如图 3-11 所示为某大桥的主桁架。主桁架的下弦杆与两个斜杆的中心线不交于一点，而是相差距离 e ，如果忽略它将导致误差。由于结点 1 和结点 2 都在刚性很大的结点块上，因此可假定它们的线位移和转角都相等。

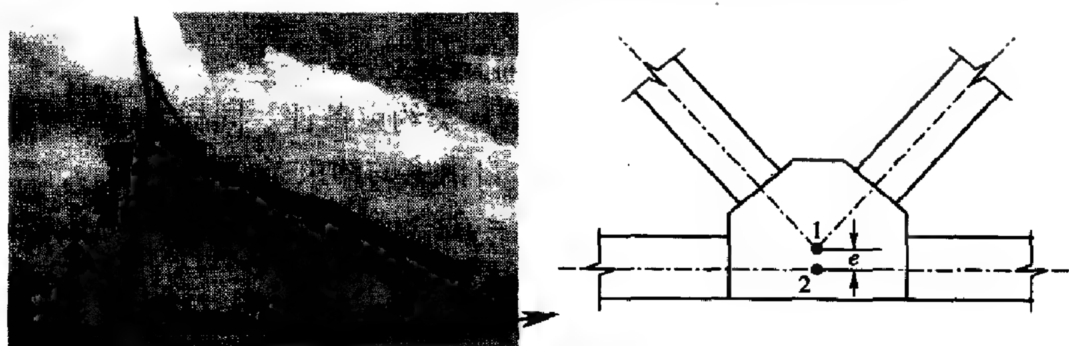


图 3-11　单元轴线不交于一点

另外，有时会遇到有铰接点的梁单元。如图 3-12 所示是一带挂梁的 T 形刚构，挂梁铰接于牛腿处。有限元模型中，有 4 个单元汇交于该结点，其中单元 4 与其铰接，其他杆件则为刚接。在该结点处，各单元具有相同的线位移，但截面转角不相同，此时可以在该处定义两个坐标一样的结点，然后指定这两个结点的线位移相等。

还有一种情形是，在悬浮体系的斜拉桥中，主梁与桥塔交界处为竖向支座，水平方向位移和转动位移未受约束，如图 3-13 所示。在离散为有限元模



图 3-12 铰接的梁单元

型时，在这里也可以设置两个结点，然后指定它们的竖直位移相等。

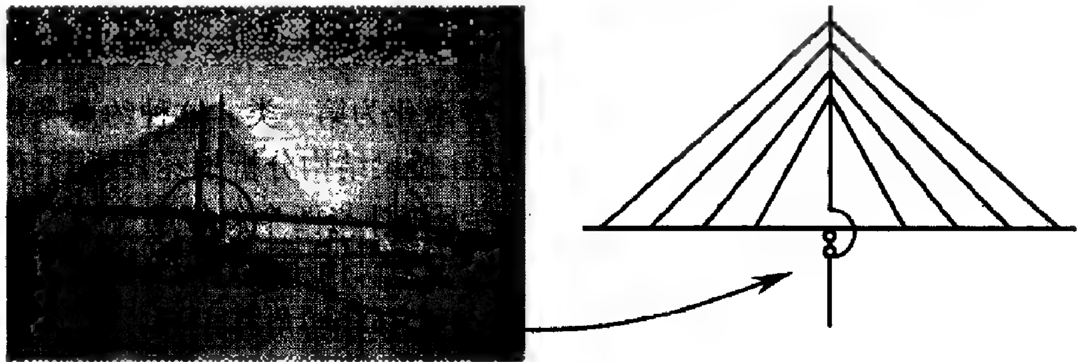


图 3-13 悬浮体系斜拉桥的梁塔交叉结点

引入第二类约束条件的方法可参照第一类约束条件。同样考虑一个只有6个方程的式(3-112)，设第2个和第5个自由度的约束条件为

$$\delta_2 + \alpha\delta_5 = \beta \tag{3-120}$$

考虑约束条件(3-120)以后，方程(3-112)可写为

$$\begin{bmatrix} k_{11} & 0 & k_{13} & k_{14} & k_{15}-\alpha k_{12} & k_{16} \\ 0 & 1 & 0 & 0 & \alpha & 0 \\ k_{31} & 0 & k_{33} & k_{34} & k_{35}-\alpha k_{32} & k_{36} \\ k_{41} & 0 & k_{43} & k_{44} & k_{45}-\alpha k_{42} & k_{46} \\ k_{51}-\alpha k_{12} & \alpha & k_{53}-\alpha k_{32} & k_{54}-\alpha k_{42} & k_{55}-2\alpha k_{52}+\alpha^2(k_{22}+1) & k_{56}-\alpha k_{62} \\ k_{61} & 0 & k_{63} & k_{64} & k_{65}-\alpha k_{62} & k_{66} \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \\ \delta_5 \\ \delta_6 \end{Bmatrix}$$

$$
= \begin{Bmatrix} f_1 - \beta k_{12} \\ \beta \\ f_3 - \beta k_{32} \\ f_4 - \beta k_{42} \\ f_5 - \alpha f_2 - \beta k_{52} + \alpha\beta k_{22} + \alpha\beta \\ f_6 - \beta k_{62} \end{Bmatrix} \tag{3-121}
$$

这样的修改保持了刚度矩阵对称带状的特性。

为了简单起见，把第二类约束条件简化成两个结点的某一个自由度相等，此时可令公式(3-120)中的 $\alpha = -1, \beta = 0$。这种情况在土木工程中非常常见，有些参考书中称之为主从结点约束，有限元软件 Ansys 中称之为自由度耦合。前面的讨论中提到的单元轴线不交于一点或铰接的梁单元都可以通过这种约束实现。其程序如下：

```
1.  [bc2_number,dummy] = size(gBC2);
2.  for ibc=1:1:bc2_number
3.      n1 = gBC2(ibc, 1 );
4.      n2 = gBC2(ibc, 2 );
5.      d = gBC2(ibc, 3 );
6.      m1 = (n1-1)*3 + d ;
7.      m2 = (n2-1)*3 + d ;
8.      f(m2) = f(m2) + f(m1) ;
9.      f(m1) = 0;
10.      gK(m2,:) = gK(m2,:) + gK(m1,:) ;
11.      gK(:,m2) = gK(:,m2) + gK(:,m1) ;
12.      gK(m1,:) = zeros(1, node_number*3) ;
13.      gK(:,m1) = zeros(node_number*3,1) ;
14.      gK(m1,m1) = 1.0 ;
15.      gK(m1,m2) = -1 ;
16.      gK(m2,m1) = -1 ;
17.      gK(m2,m2) = gK(m2,m2) + 1 ;
18.  end
```

变量 gBC2 是保存第二类边界条件的矩阵，每一行为一个边界条件，第 1 列是第 1 个结点，第 2 列是第 2 个结点，第 3 列是自由度号。其他变量与前面处理第一类边界条件程

序段相同。

如果结点荷载和约束条件已知，则由上述方法引入边界条件，然后求解联立方程组得出结点位移值。结点位移求得后，再通过各单元的平衡方程可以求出各单元在整体坐标系下的结点力 f^e，以平面梁单元为例，即

$$f^e = \begin{bmatrix} F_{xi} & F_{yi} & M_{i} & F_{xj} & F_{yj} & M_{j} \end{bmatrix}^T = K^e\delta^e \tag{3-122}$$

这里要注意的是，在 f^e 中包含了各种荷载的等效结点力。另外在工程中经常要用的剪力、轴力和弯矩是在单元的局部坐标系下定义的，因此还需要用到 **3.5** 节中的坐标变换。

在计算单元的各结点力时，需要从整体的结点位移向量中提取某个单元的结点位移向量。这里又要用到结点 n 的第 d 个自由度即为整体结点位移向量中的第 m = (n - 1) * 3 + d 个分量的概念。另外，如果单元上作用有非结点荷载，那么根据单元平衡方程得到的结点力中包含了这些非结点荷载的等效结点力，减去它们才是真正的单元反力。最后一个是坐标转换的问题，我们既可以给出整体坐标系下的单元结点力，也可以给出局部坐标系下的单元结点力。土木工程中经常要用的剪力、轴力和弯矩，是在单元的局部坐标系下进行定义的。注意到上述几个问题后，就很容易用 MATLAB 编写计算单元结点力的程序了。

```
1.    function enf = ElementNodeForce( ie )
2.    %   计算单元的结点力
3.    %   输入参数
4.    %      ie  ------结点号
5.    %   返回值
6.    %      enf ------单元局部坐标系下的结点力
7.        global gElement gNode gDelta gDF
8.        i = gElement( ie, 1 ) ;
9.        j = gElement( ie, 2 ) ;
10.       de = zeros( 6, 1 ) ;
11.       de( 1:3 ) = gDelta( (i - 1)* 3 +1:(i - 1)* 3 + 3 ) ;
12.       de( 4:6 ) = gDelta( (j - 1)* 3 +1:(j - 1)* 3 + 3 ) ;
13.       k = StiffnessMatrix( ie, 1 ) ;
14.       enf = k * de ;
15.
16.       [ df_number, dummy] = size( gDF ) ;
17.       for idf = 1:1:df_number
```

```
18.        if ie == gDF( idf, 1 )
19.            enf = enf - EquivalentNodeForce( gDF(idf,1),...
20.                gDF(idf, 2), gDF(idf, 3), gDF(idf,4) ) ;
21.            break ;
22.        end
23.    end
24.
25.    T = TransformMatrix( ie ) ;
26.    enf = transpose( T ) * enf ;
27. return
```

全局变量 gElement,gNode,gDelta和 gDF分别是单元矩阵、结点矩阵、整体结点位移向量和分布力矩阵，gDF是作用在单元上的分布荷载矩阵，每一行表示一个单元上的荷载分布，它的第 1 列表示单元号，第 2 列表示单元第一个结点上的分布力集度，第 3 列表示单元第二个结点上的分布力集度，第 4 列表示自由度号，即根据自由度号分别表示分布轴力，分布横向力和分布弯矩。第 8 和第 9 行得到单元的两个结点号，第 10 行定义一个 6 行的单元结点位移列阵，第 11 和 12 行从整体结点位移向量中提取出单元的结点位移向量，第 13 行得到单元刚度矩阵，第 14 行得到了包含分布力等效结点力的单元结点力向量。第 16～23 行把分布力等效结点力从单元结点力向量中去掉，第 25 和 26 行把单元结点力从整体坐标系转换到局部坐标系。

3.8 温度应力

在前面的讨论中，我们只考虑由外力作用引起的应力和变形，其实物体的应力和变形还可能由于温度变化或温度场不均匀引起。因为通常的材料具有热胀冷缩的特性，如果这种胀缩不受任何约束，那么虽然有变形却不产生应力。反之，如果变形受到外部约束，或物体各部分温度变化不均匀而导致变形的不协调，则为了使变形后的物体能够满足约束条件并保持连续性，将在物体内引起附加的应力，这种由于温度变化引起的应力称为“热应力”或“温度应力”。例如，一跨静定简支梁产生均匀的温度改变时，由于一端设有活动支座，允许膨胀，故无热应力产生，而两端固支的梁必然要产生热应力。我国许多地区温度变化幅度大，温度变化产生的附加内力不容忽视。另外，在高温下工作的结构，如汽轮机、燃气轮机的叶轮、叶片和气缸、化工反应塔、原子能反应堆、火箭或超音速飞机等，热应力的分析显得比较重要。另外，铸件在凝固和冷却

过程中，水坝从建造到以后蓄水过程中，温度都有很大的变化，设计时必须考虑热应力的影响。

最一般的情形，温度的变化将引起变形和应力，而反过来变形也将产生热量，从而又引起温度的变化，因此两者是耦合的。这给解决热弹性问题带来很大的数学困难。不过如果变形过程中物体的体积保持不变（不可压缩材料），或变化速度非常缓慢，则变形对温度的影响将很小，可以忽略，从而变为非耦合问题。对于这类问题，在求热应力之前，可以把温度场作为一个独立的问题来处理。幸运的是，工程中大多数问题都属于这种情况。此时，可以把温度应力问题看成由于温度变化导致的初应变问题。

当物体中存在初应变 $\boldsymbol{\varepsilon}_0$时，应力应变关系为

$$\boldsymbol{\sigma} = \boldsymbol{D}(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0) \tag{3-123}$$

由于温度变化引起的初应变为

$$\boldsymbol{\varepsilon}_0 = \boldsymbol{a}T \tag{3-124}$$

式中，$\boldsymbol{a}$ 是材料的热膨胀系数向量，T是温度的变化量。对于热膨胀系数为 α 的各向同性材料，则有

$$\boldsymbol{a} = \alpha\begin{bmatrix}1 & 1 & 1 & 0 & 0 & 0\end{bmatrix}^{\mathrm{T}} \tag{3-125}$$

然后，用式(3-123)代替第2章的式(2-48)，可以把温度应变看成另一种荷载，利用最小势能原理，把其写成等效结点力，即

$$\boldsymbol{f}_T^e = \iiint_{\Omega} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{\varepsilon}_0 \mathrm{d}\Omega = \iiint_{\Omega} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{a}T\mathrm{d}\Omega \tag{3-126}$$

因此，结构温度应力问题和无热荷载的应力分析问题相比，除增加一项以初应变形式出现的温度荷载以外，则完全相同的。因此，结构温度应力的计算过程是，先通过实验或计算或其他方法得到结构的温度场，然后把温度作为热荷载进行结构的应力分析。

对于弹性模量为 E，截面积为 A，梁长为 l 的梁单元，式(3-126)可以简化为

$$\begin{bmatrix}F_{Ni} & F_{Nj}\end{bmatrix} = \begin{bmatrix}-\dfrac{EA\alpha}{l} & \dfrac{EA\alpha}{l}\end{bmatrix}\int_0^l T\mathrm{d}x \tag{3-127}$$

一般可以假设温度场沿梁长线性分布，那么上式为

$$\begin{bmatrix}F_{Ni} & F_{Nj}\end{bmatrix} = \frac{(T_i + T_j)EA\alpha}{2}\begin{bmatrix}-1 & 1\end{bmatrix} \tag{3-128}$$

式中 T_i 和 T_j 分别是梁单元两个结点处的温度变化量。

下面是计算温度荷载的等效结点力的 MATLAB程序.

```
1.  [illegible]
2.  %    计算单元的温度荷载的等效结点力
3.  %    输入参数
4.  %        ie ------结点号
5.  %    返回值
6.  %        etf ------整体坐标系下的等效结点力
7.       global gElement gNode gMaterial
8.
9.       dT1 = gNode( gElement( ie, 1 ), 3 ) ;
10.      dT2 = gNode( gElement( ie, 2 ), 3 ) ;
11.      E = gMaterial( gElement( ie, 3 ), 1 ) ;
12.      A = gMaterial( gElement( ie, 3 ), 3 ) ;
13.      alpha = gMaterial( gElement( ie, 3 ), 5 ) ;
14.      Nx = E* A* alpha ;
15.
16.      etf = [ -Nx; 0; 0; Nx; 0; 0 ] ;
17.      T = TransformMatrix( ie ) ;
18.  [illegible]
19.  [illegible]
```

3.9 算例

3.9.1 受分布力和集中力作用的刚架

计算如图 3-14 所示平面刚架各结点的位移和各梁的内力及支座反力。首先我们来建立有限元模型。所谓有限元模型，就是定义结点坐标，定义单元的结点号和材料特性，定义约束条件，给定结点力等。我们把杆件的连接点和集中力的作用点取为结点，并从 1 ~7 编号，其中 5 号结点就是集中力作用点，如图 3-14 所示。为了确定结点的坐标，我们要先建立一个整体坐标系，其原点为结点 1，水平向右为 x 轴正方向，竖直向上为 y 轴正方向。这样就能根据框架的尺寸确定 7 个结点的坐标。然后将这 7 个结点两两组合成 6 个单元，

并从①～⑥编号，亦示于图 3-14 中。该刚架又 3 个结点被固定，每个结点有 3 个自由度，因此共有 9 个自由度被约束。另外还有 1 个集中力，两个单元有分布荷载。这些信息即构成了有限元模型，可用一个函数 **PlaneFrameModel** 来指定它，如下面的程序所示。

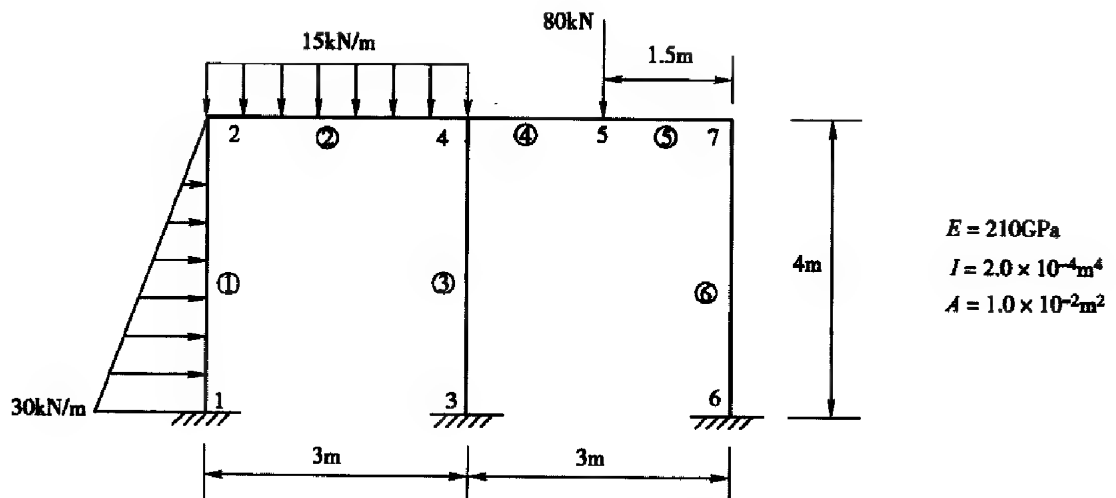


图 3-14　受分布力和集中力的平面刚架

```
1.    function PlaneFrameModel
2.    %  定义平面杆系的有限元模型
3.    %  输入参数:
4.    %      无
5.    %  返回值:
6.    %      无
7.    %  说明:
8.    %      该函数定义平面杆系的有限元模型数据:
9.    % gNode - - - - - - - 结点定义
10.   % gElement - - - - 单元定义
11.   % gMaterial - - - 材料定义,包括弹性模量,梁的截面积和梁的抗弯惯性矩
12.   % gBC - - - - - - - - - 约束条件
13.   % gNF - - - - - - - - - 集中力
14.   % gDF - - - - - - - - - 分布力
15.
16.       global gNode gElement gMaterial gBC1 gNF gDF
17.
```

```
18.         % 结点坐标
19.         %           x          y
20.         gNode = [0.0,      0.0     % 结点1
21.                  0.0,      4.0     % 结点2
22.                  3.0,      0.0     % 结点3
23.                  3.0,      4.0     % 结点4
24.                  4.5,      4.0     % 结点5
25.                  6.0,      0.0     % 结点6
26.                  6.0,      4.0 ] ;  % 结点7
27.
28.         % 单元定义
29.         %              结点1   结点2   材料号
30.         gElement = [1,       2,       1         % 单元1
31.                     2,       4,       1         % 单元2
32.                     3,       4,       1         % 单元3
33.                     4,       5,       1         % 单元4
34.                     5,       7,       1         % 单元5
35.                     6,       7,       1] ;      % 单元6
36.
37.         % 材料性质
38.         %                 弹性模量   抗弯惯性矩   截面积
39.         gMaterial = [2.1e11,    2.0e-4,     1.0e-2] ;   % 材料1
40.
41.         % 第一类约束条件
42.         %           结点号     自由度号     约束值
43.         gBC1 = [ 1,         1,          0.0
44.                  1,         2,          0.0
45.                  1,         3,          0.0
46.                  3,         1,          0.0
47.                  3,         2,          0.0
48.                  3,         3,          0.0
49.                  6,         1,          0.0
50.                  6,         2,          0.0
51.                  6,         3,          0.0] ;
52.
53.         % 集中力
```

```
54      %    结点号    [illegible]    [illegible]
55      [illegible]
56      [illegible]
57      %  [illegible]
58      %      单元号   [illegible]   [illegible]   [illegible]
59      gDF = [ 1  [illegible]  -3e3  [illegible]  2
60              2  [illegible]  -1.5e3  -1.5e3  2 ] ;
61      return
```

有了有限元模型数据，下一步的工作就是求解。其具体流程如图 **3-15** 所示。完整的程序见本书所附程序 **exam3 _ 1. m**。其计算结果如下：

图 3-15　有限元程序流程图

[illegible]	[illegible]	[illegible]	[illegible]
1	[illegible]e-019	[illegible]e-020	-2.78411083e-019
2	7.[illegible]e-004	-[illegible]e-005	-3.446[illegible]e-005
3	2.[illegible]e-019	-1.29980350e-019	-1.62999824e-019

4	[illegible]	[illegible]	[illegible]
5	7.63456283e-[illegible]	-5.67794226e-[illegible]	[illegible].15692[illegible]
6	1.29964769e-018	-7.70240078e-020	-4.104301[illegible]-019
7	7.56145765e-004	-7.70240078e-005	2.71750[illegible]e-004

		轴力	剪力	弯矩
单元号 1	结点号 1	1.63227121e+004	4.76656742e+004	3.56932655e+004
	结点号 2	-1.63227121e+004	1.23343258e+004	-5.0305[illegible]e+003
单元号 2	结点号 2	1.2[illegible]43258e+004	1.63227121e+004	5.030568[illegible]+003
	结点号 4	-1.23343258e+004	2.86772879e+004	[illegible]2.35624322e+004
单元号 3	结点号 3	6.82396838e+004	2.0[illegible]e+003	6.[illegible]599262e+003
	结点号 4	-6.82396838e+004	-2.09960018e+003	1.5524[illegible]11e+003
单元号 4	结点号 4	1.02347256e+004	3.95623959e+004	2.20[illegible]e+004
	结点号 5	-1.0234725[illegible]+004	-3.[illegible]623959e+004	3.73335696e+004
单元号 [illegible]	结点号 [illegible]	[illegible]	[illegible]	[illegible]+004
	结点号 7	-1.02347256[illegible]+004	[illegible].64376041e+004	[illegible]

3.9.2 在自重和温度荷载作用下的斜腿刚构

图3-16所示为一跨径25m的斜腿刚构桥桥型布置图，我们要分析它在自重和温度荷载下的结构变形和内力。材料特性为弹性模量 $E = 3 \times 10^{10}$ Pa，密度 $\rho = 2500$ kg/m^3，线膨胀系数 $\alpha = 1 \times 10^{-5}$。首先建立有限元模型，共51个结点，48个单元，如图3-17所示。这里需要注意桥型布置图中虚线标注处的约束条件。

由于在3.9.1算例中，有限元模型数据量较少，因此将它们全部放入源程序。这种方法的缺点是不同的计算对象都要修改源程序。因此本算例采用的是把模型数据放在外部的数据文件中，在有限元程序中从该数据文件中读取的方法。这样，不同的计算模型只要修改外部数据文件即可。

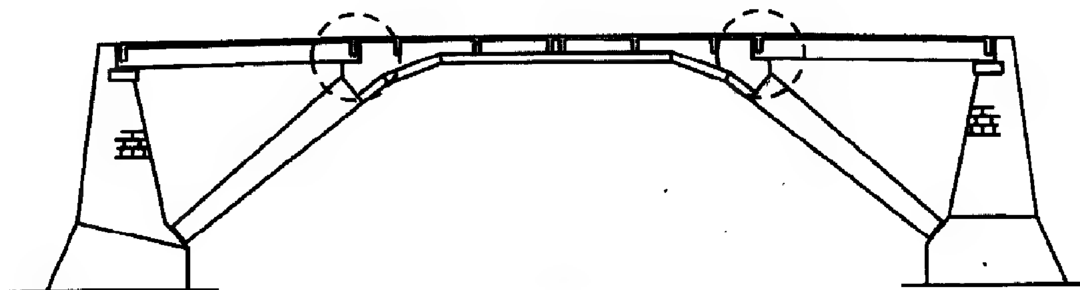


图 3-16 某桥桥梁布置图

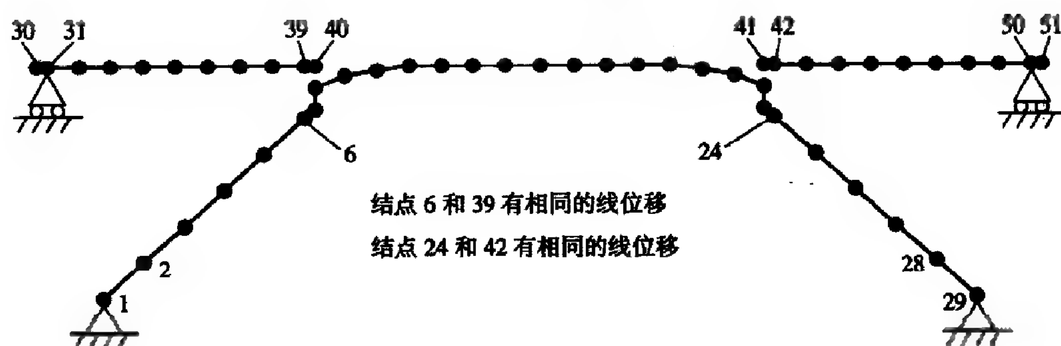


图 3-17 有限元模型

该算例的数据文件和源程序见本书所附文件　　　　**2. dat** 和 **exam3 _ 2. m**。其计算结果如下：

[illegible]	[illegible]	[illegible]	[illegible]
1	[illegible]	[illegible]	[illegible]
2	[illegible]	[illegible]	[illegible]
3	1.42716314e-004	1.27871410e-004	1.02354065e-004
4	8.34305915e-005	3.46237225e-004	1.96288058e-004
5	-4.43256735e-005	6.45875022e-004	2.55965214e-004
6	-2.06395984e-004	9.88185406e-004	2.74753818e-004
7	-2.46895847e-004	1.05802737e-003	[illegible]
8	-3.88338524e-004	1.12716432e-003	2.40274429e-004
9	[illegible]	[illegible]	[illegible]
10	-3.39991712e-004	[illegible]	[illegible]

11	-2.88014907e-004	1.61412714e-003	1.27111375e-004
12	-2.17031565e-004	1.68813886e-003	6.32399016e-005
13	-1.46048224e-004	1.72262937e-003	2.68123881e-005
14	-7.80648832e-005	1.73596348e-003	8.85268714e-006
15	6.41325847e-016	1.73933407e-003	1.88482208e-016
16	7.80648832e-005	1.73596348e-003	-8.85268714e-006
17	1.46048224e-004	1.72262937e-003	-2.68123881e-005
18	2.17031565e-004	1.68813886e-003	-6.32399076e-005
19	2.88014907e-004	1.61412714e-003	-1.21180378e-004
20	3.39991712e-004	1.48496342e-003	-1.76324386e-004
21	3.88117529e-004	1.32101627e-003	-2.04853028e-004
22	3.88338524e-004	1.12716482e-003	-2.40272471e-004
23	2.46898847e-004	1.07602737e-003	-2.71303696e-004
24	2.06395984e-004	9.88245010e-004	-2.74955810e-004
25	4.43256735e-005	6.45875022e-004	-2.55907214e-004
26	-8.84805915e-005	3.46237225e-004	-1.99208386e-004
27	-1.42271631e-004	1.27971418e-004	-1.0[illegible]
28	-1.10765129e-004	1.65442502e-005	4.46[illegible]e-006
29	0.00000000e+000	0.00000000e+000	4.70050[illegible]e-005
30	-8.71393881e-004	8.32286024e-005	-3.32801[illegible]e-005
31	-8.46393881e-004	0.00000000e+000	-3.33213390e+005
32	-7.86393881e-004	-2.23932594e-005	-1.70742971e-005
33	-6.86393881e-004	-2.88589108e-005	7.50135126e-005
34	-6.06393881e-004	-2.23648581e-005	8.52661574e-005
35	-5.26393881e-004	1.17985154e-004	1.59998678e-004
36	-4.46393881e-004	2.69404342e-004	2.23532744e-004
37	-3.66393881e-004	4.73220058e-004	2.89785[illegible]e-004
38	-2.86393881e-004	3.18725200e-004	3.28278633e-004

39	-2.06393881e-004	9.88157936e-004	3.42124674e-004
40	-1.81393881e-004	1.07368064e-003	3.42079503e-004
41	1.81393881e-004	1.07368064e-003	-3.42079503e-004
42	2.06393881e-004	9.88157936e-004	-3.42124674e-004
43	2.86393881e-004	7.18725200e-004	-3.26276631e-004
44	3.66393881e-004	4.73220058e-004	-2.83785842e-004
45	4.46393881e-004	2.69404342e-004	-2.23533218e-004
46	5.26393881e-004	1.17935154e-004	-1.54399678e-004
47	6.06393881e-004	2.23648581e-005	-8.52861174e-005
48	6.86393881e-004	-2.08589106e-005	-2.80188136e-005
49	7.66393881e-004	-2.23932924e-005	1.74772774e-005
50	8.46393881e-004	0.00000000e+000	2.83[illegible]e-005
51	8.71393881e-004	8.32286034e-006	3.3[illegible]e-005

		轴力	剪力	弯矩
单元号 1	结点号 1	1.55402739e+005	-1.14911391e+004	6.[illegible]e-012
	结点号 2	-1.51732859e+005	-7.44496408e+003	1.27823921e+004
单元号 2	结点号 2	1.51732859e+005	7.44496408e+003	-1.27823921e+004
	结点号 3	-1.47629659e+005	-2.92103908e+003	1.97797803e+004
单元号 3	结点号 3	1.47632634e+005	2.76656782e+003	-1.97797803e+004
	结点号 4	-1.43094450e+005	2.22813418e+003	2.01443170e+004
单元号 4	结点号 4	1.43096701e+005	-2.07670864e+003	-2.01443170e+004
	结点号 5	-1.38133529e+005	7.54878364e+003	1.5[illegible]e+004
单元号 5	结点号 5	1.38042346e+005	-7.69377727e+003	-1.5[illegible]e+004
	结点号 6	-1.32845[illegible]e+005	1.25[illegible]e+004	[illegible]e+002
单元号 6	结点号 6	1.16[illegible]e+005	-9.78670891e+003	1.23208463e+003
	结点号 7	-1.10696[illegible]e+005	2.1942521[illegible]e+004	-1.0929[illegible]e+004
单元号 7	结点号 7	5.18745678e+004	-3.07398407e+005	1.16004780e+004
	结点号 8	-4.79840809e+004	3.07398407e+005	-7.01471631e+004

单元号 8 结点号 6 1.17533760e+005 4.53401605e+003 7.01471531e+004

结点号 9 -1.14789340e+005 2.30424070e+003 -6.93461601e+004

单元号 9 结点号 9 1.12540371e+005 2.27281097e+004 [illegible]

结点号 10 -1.11169914e+005 -1.43585897e+004 [illegible]

单元号10 结点号 10 1.11033402e+005 1.53789891e+004 [illegible]

结点号 11 -1.09984414e+005 -8.58218747e+003 [illegible]

单元号11 结点号 11 1.07388407e+005 2.52577752e+004 [illegible]

结点号 12 -1.07388407e+005 [illegible] [illegible]

单元号12 结点号 12 1.07388407e+005 [illegible] [illegible]

结点号 13 -1.07388407e+005 [illegible] [illegible]

单元号13 结点号 13 1.07388407e+005 [illegible] [illegible]

结点号 14 -1.07388407e+005 [illegible] [illegible]

单元号14 结点号 14 1.07388407e+005 6.58289520e+003 [illegible]

结点号 15 -1.07388407e+005 [illegible] [illegible]

单元号15 结点号 15 1.07388407e+005 [illegible] [illegible]

结点号 16 -1.07388407e+005 6.58289520e+003 [illegible]

单元号16 结点号 16 1.07388407e+005 [illegible] [illegible]

结点号 17 -1.07388407e+005 1.28076552e+004 [illegible]

单元号17 结点号 17 1.07388407e+005 -1.28076552e+004 [illegible]

结点号 18 -1.07388407e+005 1.90320162e+004 [illegible]

单元号18 结点号 18 1.07388407e+005 -1.90328132e+004 [illegible]

结点号 19 -1.07388407e+005 2.52577752e+004 [illegible]

单元号19 结点号 19 1.09984414e+005 -8.58218747e+003 [illegible]

结点号 20 -1.11033402e+005 [illegible] [illegible]

单元号20 结点号 20 1.11169914e+005 [illegible] [illegible]

结点号 21 -1.12540371e+005 [illegible] [illegible]

单元号21 结点号 21 1.14789340e+005 2.30424070e+003 [illegible]

结点号 22 -1.17533760e+005 4.53401605e+003 -7.01471531e+004

单元号22	结点号 22	4.79840807e+004	1.07388407e+005	7.01471531e+004
	结点号 23	-5.15745679e+004	-1.07388407e+005	-1.16204710e+004
单元号23	结点号 23	1.15599622e+005	2.11247752e+004	9.52566897e+003
	结点号 24	-1.17011557e+005	-1.95566527e+004	-2.65553149e+003
单元号24	结点号 24	1.32429761e+005	1.36651849e+004	6.73169993e+002
	结点号 25	-1.37826252e+005	-7.71535968e+003	1.37592876e+004
单元号25	结点号 25	1.38113529e+005	7.54878364e+003	-1.36468441e+004
	结点号 26	-1.43096761e+005	-2.07670864e+003	2.01443170e+004
单元号26	结点号 26	1.42094450e+005	2.22643418e+003	-2.01443170e+004
	结点号 27	-1.47632634e+005	2.76656782e+003	1.97797303e+004
单元号27	结点号 27	1.47629659e+005	-2.92103908e+003	-1.97797303e+004
	结点号 28	-1.51732858e+005	7.44496408e+003	1.27823921e+004
单元号28	结点号 28	1.51732859e+005	-7.44496408e+003	-1.27823921e+004
	结点号 29	-1.55402739e+005	1.14911391e+004	-1.70530257e-013
单元号29	结点号 30	-3.60887458e-009	9.31095201e-011	3.67986334e-012
	结点号 31	3.60887498e-009	1.84730000e+003	-2.30912500e+002
单元号30	结点号 31	1.97906047e-009	2.36484400e+004	2.30912500e+002
	结点号 32	-1.97906047e-009	-1.77340800e+004	1.63208955e+004
单元号31	结点号 32	1.97906047e-009	1.77340800e+004	-1.63208955e+004
	结点号 33	-1.97906047e-009	-1.18227200e+004	2.81436155e+004
单元号32	结点号 33	2.91038305e-009	1.18227200e+004	-2.81436155e+004
	结点号 34	-2.91038305e-009	-5.91136000e+003	3.52372475e+004
单元号33	结点号 34	1.97906047e-009	5.91136000e+003	-3.52372475e+004
	结点号 35	-1.97906047e-009	5.66852345e-009	3.76017915e+004
单元号34	结点号 35	2.91038305e-009	-5.64295968e-009	-3.76017915e+004
	结点号 36	-2.91038305e-009	5.91136000e+003	3.52372475e+004
单元号35	结点号 36	2.91038305e-009	-5.91136000e+003	-3.52372475e+004
	结点号 37	-2.91038305e-009	1.18227200e+004	2.81436155e+004

单元号	结点号				
单元号36	结点号	37	1.97906047e-009	-1.18227200e+004	-2.81436155e+004
	结点号	38	-1.97906047e-009	1.77340800e+004	1.63208955e+004
单元号37	结点号	38	1.97906047e-009	-1.77340800e+004	-1.63208955e+004
	结点号	39	-1.97906047e-009	2.36454400e+004	-2.30912500e+002
单元号38	结点号	39	6.63567334e-009	1.84730000e+003	2.30912500e+002
	结点号	40	-6.63567334e-009	[illegible]4540e-008	5.45719080e-009
单元号39	结点号	41	1.68802217e-008	-2.[illegible]9692e-009	1.22624044e-009
	结点号	42	-1.68802217e-008	1.84730000e+003	-2.30912500e+002
单元号40	结点号	42	1.97906047e-009	2.36454400e+004	2.30912500e+002
	结点号	43	-1.97906047e-009	-1.77340800e+004	1.63208955e+004
单元号41	结点号	43	1.97906047e-009	1.77340800e+004	-1.63208955e+004
	结点号	44	-1.97906047e-009	-1.18227200e+004	2.81436155e+004
单元号42	结点号	44	2.91038305e-009	1.18227200e+004	-2.81436155e+004
	结点号	45	-2.91038305e-009	-5.91136000e+003	3.52372475e+004
单元号43	结点号	45	2.91038305e-009	5.91136000e+003	-3.52372475e+004
	结点号	46	-2.91038305e-009	6.13908924e-011	3.76017915e+004
单元号44	结点号	46	1.97906047e-009	1.05728759e-009	-3.76017915e+004
	结点号	47	-1.97906047e-009	5.91136000e+003	3.52372475e+004
单元号45	结点号	47	1.97906047e-009	-5.91136000e+003	-3.52372475e+004
	结点号	48	-1.97906047e-009	1.18227200e+004	2.81436155e+004
单元号46	结点号	48	2.91038305e-009	-1.18227200e+004	-2.81436155e+004
	结点号	49	-2.91038305e-009	1.77340800e+004	1.63208955e+004
单元号47	结点号	49	1.97906047e-009	-1.77340800e+004	-1.63208955e+004
	结点号	50	-1.97906047e-009	2.36454400e+004	-2.30912500e+002
单元号48	结点号	50	1.16415322e-010	1.84730000e+003	2.30912500e+002
	结点号	51	-1.16415322e-010	3.25940164e-010	-6.20872243e-011

参考文献

[1] 丁皓江,何福保,谢贻权,徐兴. 弹性和塑性力学中的有限单元法. 北京:机械工业出版社,1989.

[2] 刘鸿文. 材料力学(上册). 3版. 北京:高等教育出版社,1992.

[3] 《现代数学手册》编纂委员会. 现代数学手册(经典数学卷). 武汉:华中科技大学出版社,1999.

[4] Turner M. J., Clough R. W., Martin H. C., and Topp L. J.. Stiffness and deflection analysis of complex structures. Journal of Aeronautical Sciences, 1956. 23(9): 805 - 824.

[5] Martin H. C.. Introduction to matrix methods of structural analysis. McGraw-Hill. New York, 1966.

[6] Daryl L. Logan. 伍义生,吴永礼,等译. 有限元方法基础教程. 北京:电子工业出版社,2003.

[7] 艾金 J. E.. 有限元法的应用与实现,1992

第 4 章 平面问题

几乎所有的有限元专著都把 Clough[1] 在 1960 发表的用有限元法求解平面应力问题的文章作为有限元法发明的一个重要标志。而平面问题也是弹性力学中最简单的一种情形，通过对平面问题的有限元求解，可以更加深刻地理解和掌握有限元法的本质，为用有限元法求解更复杂的问题打下扎实的基础。本章介绍两种最简单的三角形和矩形平面单元。

4.1 两类平面问题

弹性力学研究的问题可以分为空间及平面两类。严格地说，任何弹性物体都处在三维受力状态，因而都是空间问题。但是在一定条件下，许多空间问题可以简化为平面问题，从而可大大减少计算工作量。一般我们把平面问题分为平面应力和平面应变两类。下面就简要地介绍这两类平面问题的基本概念，详细可参考弹性力学的专著[2]。

4.1.1 平面应力问题

如图 4-1 所示，设有一带孔带倒角的薄板，其厚度方向的尺寸比其他两个方向的尺寸小许多，在板边上受有平行于板面并沿板厚均匀分布的荷载。则可以认为板内各点的 6 个应力分量中的 σ_z、τ_{xz} 和 τ_{yz} 均为零，而剩下的 3 个应力分量 σ_x、σ_y 和 τ_{xy} 都是平行于 xy 平面的，而且可以进一步假设它们沿板厚均匀分布，所以这种问题被称为平面应力问题。

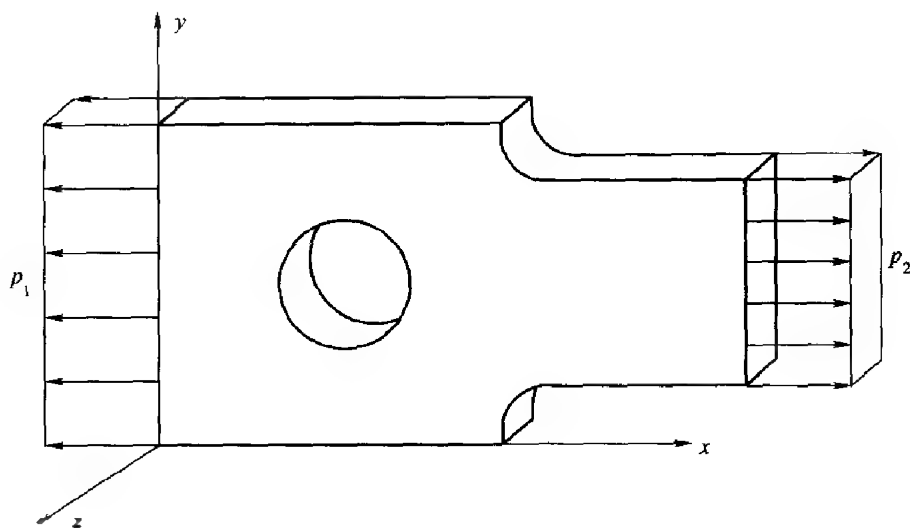


图 4-1　平面应力问题

4.1.2　平面应变问题

如图 4-2所示，设有一很长的柱体，其轴线为 z 轴，在柱面上受有平行于横截面而且沿 z 轴均匀分布的荷载 p，则可认为柱体内各点只有 x 及 y 方向的位移，而没有沿 z 轴的位移，以及应变分量 ε_z、γ_{xz} 和 γ_{yz} 均为零，剩下的 3 个应变分量 ε_x、ε_y 和 γ_{xy} 平行于 xy 平面，且与坐标 z 无关，所以这种问题称为平面应变问题。工程中受侧向土压力的挡土墙，受水压的重力坝，以及地下的管道或隧道均都可以假设为平面应变状态。

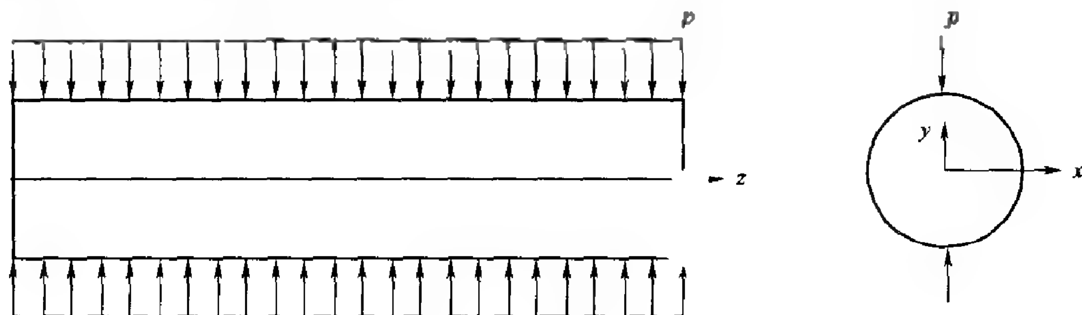


图 4-2　平面应变问题

上述两类问题合称为平面问题。应该指出，在平面应力问题中，虽然 $\sigma_z = 0$，但是 $\varepsilon_z \neq 0$；而平面应变问题中，虽然 $\varepsilon_z = 0$，但是 $\sigma_z \neq 0$。因为柱体中各点的 ε_z 之所以等于零，是受了相邻点的约束的结果，自然 σ_z 也就不会为零。

平面问题中的基本物理量为位移 $\boldsymbol{u} = [u \quad v]^{\mathrm{T}}$，应力 $\boldsymbol{\sigma} = [\sigma_x \quad \sigma_y \quad \tau_{xy}]^{\mathrm{T}}$ 和应变 $\boldsymbol{\varepsilon} = [\varepsilon_x \quad \varepsilon_y \quad \gamma_{xy}]^{\mathrm{T}}$。几何关系可以简化为

$$\boldsymbol{\varepsilon}=\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix}=\begin{bmatrix}\partial/\partial x & 0\\ 0 & \partial/\partial y\\ \partial/\partial y & \partial/\partial x\end{bmatrix}\begin{Bmatrix}u\\ v\end{Bmatrix} \tag{4-1}$$

而应力应变关系为

$$\boldsymbol{\sigma}=\boldsymbol{D\varepsilon} \tag{4-2}$$

这里要注意的是，平面应力问题和平面应变问题的弹性矩阵 $\boldsymbol{D}$ 是不同的，即对于平面应力问题为

$$\boldsymbol{D}=\frac{E}{1-\mu^2}\begin{bmatrix}1 & \mu & 0\\ \mu & 1 & 0\\ 0 & 0 & (1-\mu)/2\end{bmatrix} \tag{4-3}$$

而对于平面应变问题，则为

$$\boldsymbol{D}=\frac{E}{(1+\mu)(1-2\mu)}\begin{bmatrix}1-\mu & \mu & 0\\ \mu & 1-\mu & 0\\ 0 & 0 & (1-2\mu)/2\end{bmatrix} \tag{4-4}$$

4.2 三角形常应变单元

4.2.1 平面问题结构的离散化

有限单元法应用于结构分析时，第一步就是把结构离散化。对于平面问题，结构离散化不像杆系结构那样直接。我们可以采用很多种单元形式进行离散，但其中 3 结点的三角形单元是最早提出的，并且至今仍广泛应用的单元，图 4-3 所示为一个离散化的例子，一个受载的悬臂梁和用三角形单元离散化的模型。外荷载通过一定的规则移至结点上，这样就得到了有限元法的计算模型。

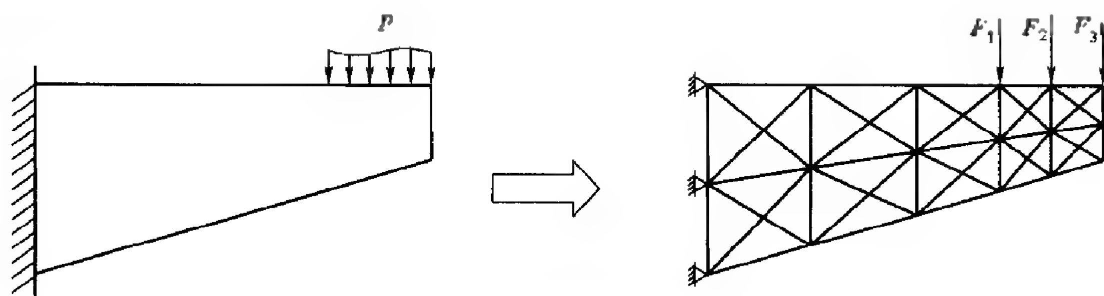


图 4-3 悬臂梁的离散化

4.2.2 位移模式和形函数的构造

首先，要建立单元结点位移与单元内任意一点位移之间的关系。图 4-4 所示为一个典型的三角形单元,其结点 i、j 和 m 按逆时针排列。每个结点位移在单元平面内有两个分量。整个单元将有 6 个结点位移分量,可用列阵表示为

$$\boldsymbol{\delta}^e = \begin{bmatrix} \boldsymbol{\delta}_i^{\mathrm{T}} & \boldsymbol{\delta}_j^{\mathrm{T}} & \boldsymbol{\delta}_m^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} u_i & v_i & u_j & v_j & u_m & v_m \end{bmatrix}^{\mathrm{T}} \tag{4-5}$$

其中子矩阵

$$\boldsymbol{\delta}_i = \begin{bmatrix} u_i & v_i \end{bmatrix}^{\mathrm{T}} \qquad (i,j,m) \tag{4-6}$$

式中 (i,j,m) 表示其他结点位移子矩阵可以按下标 i、j 和 m 的轮换得到。而 u_i 和 v_i 则是结点 i 沿 x 轴和 y 轴的位移分量。

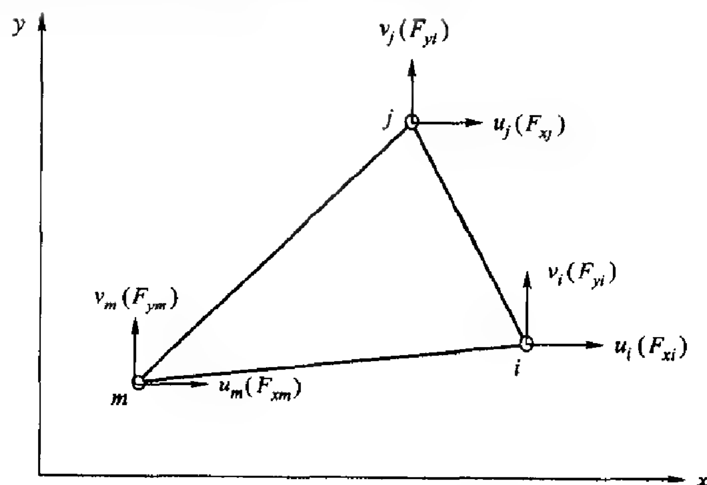


图 4-4 一个典型的三角形单元

由于单元体本身也是一个二维的弹性体，单元内各点的位移分量是坐标 x 和 y 的函数。在进行单元分析时，为了要用结点位移作为基本未知量，并用它来表示单元内任意一点的位移、变形和应力，需要假定一个位移模式。在有限元法中，位移模式一般采用多项式作为近似函数，因为它运算简便，并且随着项数的增多，可以逼近任何一段光滑的函数。这里选择最简单的线性函数作为位移模式,即

$$u = \alpha_1 + \alpha_2 x + \alpha_3 y, \quad v = \alpha_4 + \alpha_5 x + \alpha_6 y \tag{4-7}$$

式中，α_1、α_2、……、α_6 是待定常数，也称为广义坐标，可以由单元的 6 个结点位移来确定。设结点 i、j 和 m 的坐标分别为 (x_i,y_i)、(x_j,y_j) 和 (x_m,y_m)，将它们代入式(4-7)，得到

$$u_i = \alpha_1 + \alpha_2 x_i + \alpha_3 y_i \qquad v_i = \alpha_4 + \alpha_5 x_i + \alpha_6 y_i$$

$$u_j = \alpha_1 + \alpha_2 x_j + \alpha_3 y_j \qquad v_j = \alpha_4 + \alpha_5 x_j + \alpha_6 y_j$$

$$u_m = \alpha_1 + \alpha_2 x_m + \alpha_3 y_m \qquad v_m = \alpha_4 + \alpha_5 x_m + \alpha_6 y_m \tag{4-8}$$

从上式左边的 3 个方程,可得

$$\alpha_1 = \frac{1}{2\Delta}\begin{vmatrix} u_i & x_i & y_i \\ u_j & x_j & y_j \\ u_m & x_m & y_m \end{vmatrix} \qquad \alpha_2 = \frac{1}{2\Delta}\begin{vmatrix} 1 & u_i & y_i \\ 1 & u_j & y_j \\ 1 & u_m & y_m \end{vmatrix} \qquad \alpha_3 = \frac{1}{2\Delta}\begin{vmatrix} 1 & x_i & u_i \\ 1 & x_j & u_j \\ 1 & x_m & u_m \end{vmatrix} \tag{4-9}$$

式中

$$2\Delta = \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{vmatrix} \tag{4-10}$$

从解析几何[13]可知，当约定结点 i、j 和 m 的次序是逆时针转向时，如图 4-4所示,式(4-10)中的 Δ 等于三角形 ijm 的面积。当然，如果在应用式(4-10)求三角形的面积时取其绝对值,那么 3 个结点的顺序就可以不遵守这个约定。

将式(4-9)代入式(4-7)得到

$$u = \frac{1}{2\Delta}\left[(a_i + b_i x + c_i y)u_i + (a_j + b_j x + c_j y)u_j + (a_m + b_m x + c_m y)u_m\right] \tag{4-11}$$

式中

$$\left.\begin{aligned} a_i &= x_j y_m - x_m y_j \\ b_i &= y_j - y_m \qquad (i,j,m) \\ c_i &= -(x_j - x_m) \end{aligned}\right\} \tag{4-12}$$

同理得到 y 方向的位移为

$$v = \frac{1}{2\Delta}\left[(a_i + b_i x + c_i y)v_i + (a_j + b_j x + c_j y)v_j + (a_m + b_m x + c_m y)v_m\right] \tag{4-13}$$

如令

$$N_i = \frac{1}{2\Delta}(a_i + b_i x + c_i y) \qquad (i,j,m) \tag{4-14}$$

位移模式公式 (4-7)可以写成

$$u = N_i u_i + N_j u_j + N_m u_m \qquad v = N_i v_i + N_j v_j + N_m v_m \tag{4-15}$$

式中 N_i、N_j 和 N_m 是坐标的函数，它们反映单元的位移状态，因而称为形函数。

也可以把式(4-15)改写成矩阵形式

$$u = \begin{Bmatrix} u \\ v \end{Bmatrix} = N\delta^e, \quad N = \begin{bmatrix} N_i & N_j & N_m \end{bmatrix}, \quad N_i = N_i I_2 \quad (i,j,m) \tag{4-16}$$

式中,N 称为形函数矩阵。

根据式(4-11)和式(4-13),在单元的边界上位移是线性变化的,两个相邻的单元在其公共结点上具有相同的结点位移,因而在它们的公共边界上,两个单元将具有相同的位移,也就是说所选的位移函数保证了相邻单元之间位移的连续性。

4.2.3 单元应变

有了单元的位移模式,就可以利用平面问题的几何方程

$$\varepsilon = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \partial u/\partial x \\ \partial v/\partial y \\ \partial u/\partial y + \partial v/\partial x \end{Bmatrix} \tag{4-17}$$

求得应变分量。将式(4-16)代入上式得到

$$\varepsilon = \frac{1}{2\Delta}\begin{bmatrix} b_i & 0 & b_j & 0 & b_m & 0 \\ 0 & c_i & 0 & c_j & 0 & c_m \\ c_i & b_i & c_j & b_j & c_m & b_m \end{bmatrix}\begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_m \\ v_m \end{Bmatrix} = B\delta^e \tag{4-18}$$

式中的矩阵 B 可以写成分块形式

$$B = \begin{bmatrix} B_i & B_j & B_m \end{bmatrix} \tag{4-19}$$

而子矩阵

$$B_i = \frac{1}{2\Delta}\begin{bmatrix} b_i & 0 \\ 0 & c_i \\ c_i & b_i \end{bmatrix} \quad (i,j,m) \tag{4-20}$$

公式(4-18)是用结点位移表示单元应变的矩阵方程,矩阵 B 就是单元应变矩阵。由于 Δ 和 b_i、b_j、b_m、c_i、c_j 和 c_m 等都是常数,所以矩阵 B 中的元素都是常量,因而单元中任意一点的应变分量 ε_x、ε_y 和 γ_{xy} 也都是常量,故通常称这种单元为常应变单元。

4.2.4 单元应力

在得到单元应变之后，再利用应力应变关系式（4-2)便可以得到以结点位移表示的应力。把式(4-18)代入上式,得到

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{B}\boldsymbol{\delta}^e = \boldsymbol{S}\boldsymbol{\delta}^e \tag{4-21}$$

这就是应力与结点位移的关系式,其中 $\boldsymbol{S}$ 称为应力矩阵，它可写成分块形式

$$\boldsymbol{S} = \boldsymbol{D}\begin{bmatrix} \boldsymbol{B}_i & \boldsymbol{B}_j & \boldsymbol{B}_m \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}_i & \boldsymbol{S}_j & \boldsymbol{S}_m \end{bmatrix} \tag{4-22}$$

对于平面应力问题，弹性矩阵 $\boldsymbol{D}$ 为式(4-3)，则 $\boldsymbol{S}$ 的子矩阵可以写成

$$\boldsymbol{S}_i = \frac{E}{2(1-\mu^2)\Delta}\begin{bmatrix} b_i & \mu c_i \\ \mu b_i & c_i \\ \frac{1-\mu}{2}c_i & \frac{1-\mu}{2}b_i \end{bmatrix} \quad (i,j,m) \tag{4-23}$$

对于平面应变问题，则要应用式(4-4)中的弹性矩阵 $\boldsymbol{D}$,得到

$$\boldsymbol{S}_i = \frac{E(1-\mu)}{2(1+\mu)(1-2\mu)\Delta}\begin{bmatrix} b_i & \frac{\mu}{1-\mu}c_i \\ \frac{\mu}{1-\mu}b_i & c_i \\ \frac{1-2\mu}{2(1-\mu)}c_i & \frac{1-2\mu}{2(1-\mu)}b_i \end{bmatrix} \quad (i,j,m) \tag{4-24}$$

注意到式(4-5)，式(4-21)可以写成

$$\boldsymbol{\sigma} = \boldsymbol{S}_i\boldsymbol{\delta}_i + \boldsymbol{S}_j\boldsymbol{\delta}_j + \boldsymbol{S}_m\boldsymbol{\delta}_m \tag{4-25}$$

从式(4-23)或式(4-24)可以看出,$\boldsymbol{S}$ 中的元素也都是常量，所以每个单元中的应力分量是常量。但是不同的单元将具有不同的应力和应变。这样，越过公共边界，从一个单元到另一个与它相邻的单元，应力和应变的值都将突变，但是位移是连续的。上述常应变单元的这些性质，实际上都是由所选取的线性位移模式所造成的。

4.3 形函数的性质与面积坐标

三角形单元的形函数有一些非常有用的特性，它们将在后面几节中应用，因此在这里先进行介绍，同时引入面积坐标的概念。

4.3.1 形函数的性质

在讨论常应变三角形单元时，曾提出形函数

$$N_i = \frac{1}{2\Delta}(a_i + b_i x + c_i y) \qquad (i,j,m) \tag{4-26}$$

由式(4-12)可知，常数 a_i、a_j、a_m、b_i、b_j、b_m、c_i、c_j 和 c_m 依次是行列式 2Δ 的第1行、第2行和第3行各元素的代数余子式。根据行列式的性质，行列式的任一行（列）的元素与其相应的代数余子式的乘积之和等于行列式的值，而任一行(列)的元素与其他行(列)元素的代数余子式乘积之和等于零，从而可推出形函数的许多性质如下：

(1)形函数 N_i 在结点 i 上的值

$$N_i(x_i, y_i) = \frac{1}{2\Delta}(a_i + b_i x_i + c_i y_i) = 1 \tag{4-27}$$

而在其他两个结点 j 和 m 上的值

$$N_i(x_j, y_j) = \frac{1}{2\Delta}(a_i + b_i x_j + c_i y_j) = 0 \tag{4-28}$$

$$N_i(x_m, y_m) = \frac{1}{2\Delta}(a_i + b_i x_m + c_i y_m) = 0 \tag{4-29}$$

这个性质也可以从式(4-15)得到验证。即在结点 i，应该有 $u = u_i$，因此必然要求 $N_i = 1, N_j = N_m = 0$。其他两个形函数也有同样的性质，即

$$N_i(x_j, y_j) = \begin{cases} 1 & 当 j = i \\ 0 & 当 j \neq i \end{cases} \qquad (i,j,m) \tag{4-30}$$

(2)在单元任一点上3个形函数之和等于1。可以证明如下：

$$\begin{aligned} & N_i(x,y) + N_j(x,y) + N_m(x,y) \\ =& \frac{1}{2\Delta}(a_i + b_i x + c_i y + a_j + b_j x + c_j y + a_m + b_m x + c_m y) \\ =& \frac{1}{2\Delta}\left[(a_i + a_j + a_m) + (b_i + b_j + b_m)x + (c_i + c_j + c_m)y\right] \end{aligned} \tag{4-31}$$

根据前述行列式的性质，上式右端第一个圆括号等于 2Δ，第2、3个圆括号都等于零，故有

$$N_i(x,y) + N_j(x,y) + N_m(x,y) = 1 \tag{4-32}$$

由此可见，3个形函数中只有两个是独立的。

(3)三角形单元 ijm 的 ij 边的方程可用形函数表示为

$$N_i(x,y) = 1 - \frac{x - x_i}{x_j - x_i}, \quad N_j(x,y) = \frac{x - x_i}{x_j - x_i}, \quad N_m(x,y) = 0 \tag{4-33}$$

只要把 ij 边的方程式

$$y = \frac{y_j - y_i}{x_j - x_i}(x - x_i) + y_i = -\frac{b_m}{c_m}(x - x_i) + y_i \tag{4-34}$$

代入式(4-14)中就能验证。另外两条边的方程可类似得到。

利用这一性质，可以很容易地证明单元的位移，进行线性插值之后，在公共边上是连续的。由式(4-33)可知，在公共边界 ij 上，形函数 N_i 和 N_j 的值与第 3 个结点无关，而第 3 个形函数的值为零，所以不论按照哪个单元来计算公共边上的位移，均得到

$$u = N_i u_i + N_j u_j, \quad v = N_i v_i + N_j v_j \tag{4-35}$$

所以相邻单元的位移是连续的。

4.3.2 面积坐标

现在我们引进面积坐标的概念。如图 4-5 所示，在三角形单元中，任意点 $p(x,y)$ 的位置，可以用下列三个比值来确定

$$L_i = \frac{\Delta_i}{\Delta}, \quad L_j = \frac{\Delta_j}{\Delta}, \quad L_m = \frac{\Delta_m}{\Delta} \tag{4-36}$$

式中，Δ 为单元 ijm 的面积，Δ_i、Δ_j 和 Δ_m 分别是三角形 pjm、pmi 和 pij 的面积，3 个比值 L_i、L_j 和 L_m 称为 p 点的面积坐标。显然，3 个面积坐标并不全是独立的。由于 $\Delta_i + \Delta_j + \Delta_m = \Delta$ 它们之间存在关系式

$$L_i + L_j + L_m = 1 \tag{4-37}$$

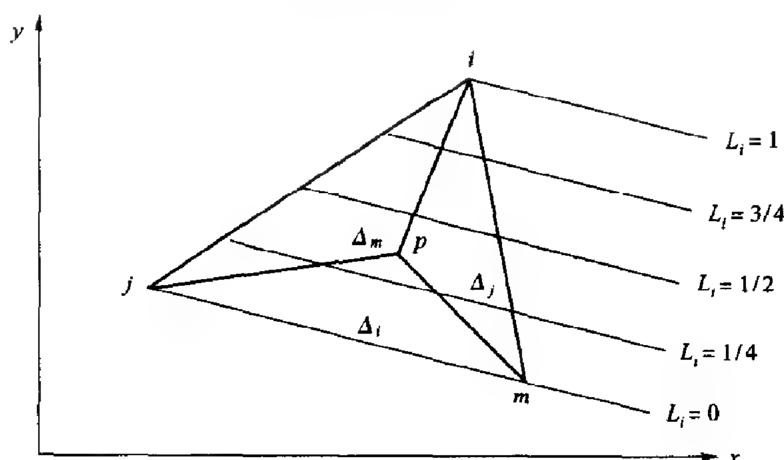


图 4-5　面积坐标

根据面积坐标的定义，不难从图 4-5 看出，与 jm 平行的每根直线上的点，都有相同的 L_i 坐标，并且等于"该直线至 jm 边的垂直距离"与"结点 i 至 jm 边垂直距离"的比值。图 4-5 中给出了 L_i 的一些等值线。容易看出，3 个结点的

面积坐标分别为

结点 i: $L_i=1, L_j=L_m=0$

结点 j: $L_j=1, L_i=L_m=0$

结点 m: $L_m=1, L_j=L_i=0$

现在再来推导面积坐标和直角坐标之间的关系。三角形 pjm 的面积是

$$\Delta_i=\frac{1}{2}\begin{vmatrix}1 & x & y\\ 1 & x_j & y_j\\ 1 & x_m & y_m\end{vmatrix}=\frac{1}{2}(a_i+b_i x+c_i y) \tag{4-38}$$

于是面积坐标

$$L_i=\frac{\Delta_i}{\Delta}=\frac{1}{2\Delta}(a_i+b_i x+c_i y) \tag{4-39}$$

类似地有

$$L_j=\frac{\Delta_j}{\Delta}=\frac{1}{2\Delta}(a_j+b_j x+c_j y) \tag{4-40}$$

$$L_m=\frac{\Delta_m}{\Delta}=\frac{1}{2\Delta}(a_m+b_m x+c_m y) \tag{4-41}$$

将式(4-39)~式(4-41)与式(4-14)对比，很容易发现前述三角形常应变单元中的形函数 N_i、N_j 和 N_m 就是面积坐标 L_i、L_j 和 L_m。

直角坐标与面积坐标有如下关系：

$$\begin{aligned}x&=x_i L_i+x_j L_j+x_m L_m\\ y&=y_i L_i+y_j L_j+y_m L_m\end{aligned} \tag{4-42}$$

证明如下：

$$\begin{aligned}x_i L_i+x_j L_j+x_m L_m&=\frac{1}{2\Delta}\Big[a_i x_i+a_j x_j+a_m x_m+(b_i x_i+b_j x_j+b_m x_m)x\\&\quad+(c_i x_i+c_j x_j+c_m x_m)y\Big]\\&=\frac{1}{2\Delta}\Big[0+2\Delta x+0y\Big]=x\end{aligned} \tag{4-43}$$

对于 y 可有类似的证明。

式(4-39)~式(4-41)和式(4-42)是面积坐标与直角坐标之间的变换公式。设 L_i 和 L_j 是独立变量，则 $L_m=1-L_i-L_j$，变换式(4-42)可以把 xy 平面上的任意三角形 ijm，变换为 $L_i L_j$ 平面上的三角形 $i_1 j_1 m_1$，如图 4-6所示。

当面积坐标的函数对直角坐标求导时，可以应用下列公式

$$
\left.\begin{aligned}
\frac{\partial}{\partial x} &= \frac{\partial L_i}{\partial x}\frac{\partial}{\partial L_i} + \frac{\partial L_j}{\partial x}\frac{\partial}{\partial L_j} + \frac{\partial L_m}{\partial x}\frac{\partial}{\partial L_m} = \frac{b_i}{2\Delta}\frac{\partial}{\partial L_i} + \frac{b_j}{2\Delta}\frac{\partial}{\partial L_j} + \frac{b_m}{2\Delta}\frac{\partial}{\partial L_m} \\
\frac{\partial}{\partial y} &= \frac{\partial L_i}{\partial y}\frac{\partial}{\partial L_i} + \frac{\partial L_j}{\partial y}\frac{\partial}{\partial L_j} + \frac{\partial L_m}{\partial y}\frac{\partial}{\partial L_m} = \frac{c_i}{2\Delta}\frac{\partial}{\partial L_i} + \frac{c_j}{2\Delta}\frac{\partial}{\partial L_j} + \frac{c_m}{2\Delta}\frac{\partial}{\partial L_m}
\end{aligned}\right\} \tag{4-44}
$$

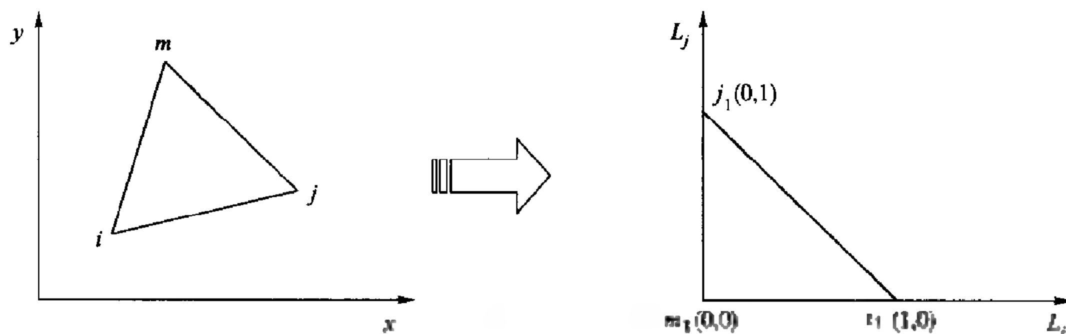


图 4-6 xy 平面上的任意三角形 ijm 变换为 L_iL_j 平面上的三角形 $i_1j_1m_1$

求面积坐标的幂函数在三角形单元上的积分时，可以应用如下积分公式

$$
\iint_{\Delta} L_i^{\alpha} L_j^{\beta} L_m^{\gamma} \mathrm{d}x\mathrm{d}y = \frac{\alpha!\beta!\gamma!}{(\alpha+\beta+\gamma+2)!}2\Delta \tag{4-45}
$$

式中，α、β 和 γ 为整常数。求面积坐标的幂函数在三角形某一边上的积分值时，可以应用如下的积分公式

$$
\int_{l} L_i^{\alpha} L_j^{\beta} \mathrm{d}s = \frac{\alpha!\beta!}{(\alpha+\beta+1)!}l \qquad (i,j,m) \tag{4-46}
$$

式中 l 为该边的长度。上面两个积分公式的证明要用到如下的积分公式

$$
\int_0^d x^{\alpha}(d-x)^{\beta}\mathrm{d}x = \frac{\alpha!\beta!}{(\alpha+\beta+1)!}d^{\alpha+\beta+1} \tag{4-47}
$$

4.4 单元刚度矩阵

根据单元刚度矩阵的一般形式（2-70），我们得到三角形单元的单元刚度矩阵为

$$
\boldsymbol{K}^e = \iiint_{\Omega} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}\mathrm{d}\Omega = \iint_{\Delta} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}h\mathrm{d}x\mathrm{d}y \tag{4-48}
$$

式中 h 是单元的厚度。根据前面的推导，应变矩阵是常数矩阵，如果单元厚度 h 也是常数，那么上式可以简化为

$$
\boldsymbol{K}^e = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}h\Delta \tag{4-49}
$$

与第 3 章中梁单元的情况相类似，单元刚度矩阵 $\boldsymbol{K}^e$ 中的元素称为刚度系

数，每一列刚度系数分别等于该单元的某个结点沿坐标轴方向发生单位位移时，在各结点上所引起的结点力。它决定于该单元的形状、大小、方位和弹性模量，而与单元的位置无关，即不随单元或坐标轴的平移而改变。

将应变矩阵的表达式(4-19)和平面应力问题的弹性矩阵(4-3)代入式(4-49)，即得平面应力问题中常应变三角形单元的刚度矩阵的显式

$$\boldsymbol{K}^e = \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ij} & \boldsymbol{K}_{im} \\ \boldsymbol{K}_{ji} & \boldsymbol{K}_{jj} & \boldsymbol{K}_{jm} \\ \boldsymbol{K}_{mi} & \boldsymbol{K}_{mj} & \boldsymbol{K}_{mm} \end{bmatrix} \tag{4-50}$$

式中

$$\boldsymbol{K}_{rs} = \boldsymbol{B}_r^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}_s h\Delta = \frac{Eh}{4(1-\mu^2)\Delta}\begin{bmatrix} b_r b_s + \frac{1-\mu}{2}c_r c_s & \mu b_r c_s + \frac{1-\mu}{2}c_r b_s \\ \mu c_r b_s + \frac{1-\mu}{2}c_s b_r & c_r c_s + \frac{1-\mu}{2}b_r b_s \end{bmatrix}$$

$$(r,s = i,j,m) \tag{4-51}$$

对于平面应变问题，只要把 E 换成 $E/(1-\mu^2)$，μ 换成 $\mu/(1-\mu)$ 即可。

我们可以利用 **MATLAB** 的符号运算功能来检验上述公式的正确性，其程序如下：

```
syms br bs cr cs E u h delta
D=E/(1-u^2)*[1        u        0
             u        1        0
             0        0        (1-u)/2];
Br=1/2/delta*[br      0
             0        cr
             cr       br];
Bs=1/2/delta*[bs      0
             0        cs
             cs       bs];
krs=transpose(Br)*D*Bs*delta*h;
krs=simple(krs);
```

在 **MATLAB** 命令窗口中执行之后，得到：

```
krs =
   E*h/8/(1-u^2)/delta*
[2*br*bs+cr*cs-cr*cs*u, 2*u*br*cs+cr*bs-cr*bs*u]
[2*u*cr*bs+br*cs-br*u*cs, 2*cr*cs+br*bs-br*bs*u]
```

很容易发现上面的 krs 即为公式(4-51)中的 K_{rs}，采用分块矩阵的写法，在公式表达上十分简洁明了，但是在实际程序的编制中，由于 MATLAB 的矩阵运算功能十分强大，可以直接采用整个矩阵相乘。下面给出了三角形平面应变单元的单元刚度矩阵计算函数：

```
1.  function k = StiffnessMatrix( ie )
2.  %   计算单元刚度矩阵
3.  %   输入参数：
4.  %      ie ---- 单元号
5.  %   返回值：
6.  %      k ---- 单元刚度矩阵
7.
8.      global gNode gElement gMaterial
9.      k = zeros( 6, 6 ) ;
10.     E = gMaterial( gElement(ie, 4), 1 ) ;
11.     mu = gMaterial( gElement(ie, 4), 2 ) ;
12.     h = gMaterial( gElement(ie, 4), 3 ) ;
13.     xi = gNode( gElement( ie, 1 ), 1 ) ;
14.     yi = gNode( gElement( ie, 1 ), 2 ) ;
15.     xj = gNode( gElement( ie, 2 ), 1 ) ;
16.     yj = gNode( gElement( ie, 2 ), 2 ) ;
17.     xm = gNode( gElement( ie, 3 ), 1 ) ;
18.     ym = gNode( gElement( ie, 3 ), 2 ) ;
19.     ai = xj* ym - xm* yj ;
20.     aj = xm* yi - xi* ym ;
21.     am = xi* yj - xj* yi ;
22.     bi = yj - ym ;
23.     bj = ym - yi ;
24.     bm = yi - yj ;
25.     ci = - (xj - xm) ;
26.     cj = - (xm - xi) ;
27.     cm = - (xi - xj) ;
28.     area = (ai + aj + am)/2 ;
29.     B = [ bi  0  bj  0  bm  0
30.           0  ci  0  cj  0  cm
31.           ci  bi  cj  bj  cm  bm] ;
32.     B = B/2/area ;
33.     D = [1 - mu  mu        0
```

```
34 [illegible]
35 [illegible]
36 [illegible]
37     k= transpose(B)* D* B* h*abs(area) [illegible]
38 return
```

第 1 行定义函数 StiffnessMatrix，输入参数 ie 和返回值变量 k，它们的含义见第 2 ~ 7 行的注释，第 8 行定义了 3 个全局变量，它们的含义与第 3 章给出的程序段一样，分别是单元矩阵、结点矩阵和材料矩阵。第 10 ~ 12 行从材料矩阵中得到了单元材料的弹性模量 E，泊松比 mu 和单元厚度 h。第 13 ~ 18 行得到了单元 3 个结点 i，j 和 m 的坐标。第 19 ~ 27 行根据公式(4-12)计算了参数 ai，aj，…，cm 的值。第 28 行计算了单元的面积 area，这里取了绝对值，因此单元结点的顺序不一定要按逆时针排列。第 29 ~ 32 行根据公式(4-20)计算了应变矩阵 B。第 33 ~ 36 行是平面应力状态的弹性矩阵 D。第 37 行是根据公式(4-49)计算了单元刚度矩阵，并把它存放到返回值变量 k 中。由于平面三角形单元在推导公式时采用整体坐标系，因此这里不像杆系单元，需要进行坐标转换。对于平面应变单元的刚度矩阵，只需修改程序中的弹性矩阵 D 即可。

4.5 等效结点力

4.5.1 单元自重

设单元的厚度为 h，单元体积密度为 ρ，自重沿 y 轴负向，因此体积力向量可写成

$$p_v = \begin{Bmatrix} 0 \\ -\rho g \end{Bmatrix} \tag{4-52}$$

式中 g 为重力加速度。将上式和形函数的定义式(4-16)代入体积力的等效结点力公式(2-72)，则可得到三角形单元的自重分配到结点上的等效结点力为

$$f_v^e = \begin{Bmatrix} f_{vi}^e \\ f_{vj}^e \\ f_{vm}^e \end{Bmatrix} = \iint_{\Delta} \begin{bmatrix} N_i^{\mathrm{T}} \\ N_j^{\mathrm{T}} \\ N_m^{\mathrm{T}} \end{bmatrix} p_v \, h \mathrm{d}x \mathrm{d}y \tag{4-53}$$

式中

$$f_{vi}^e = \iint_{\Delta} N_i \, p_v \, h \mathrm{d}x \mathrm{d}y \quad (i, j, m) \tag{4-54}$$

则结点 i 的等效结点力为

$$f_{vi}^e = \begin{Bmatrix} f_{vxi} \\ f_{vyi} \end{Bmatrix} = \iint_\Delta N_i \begin{Bmatrix} 0 \\ -\rho g \end{Bmatrix} h\mathrm{d}x\mathrm{d}y = \begin{Bmatrix} 0 \\ -\rho g h \iint_\Delta N_i \mathrm{d}x\mathrm{d}y \end{Bmatrix} \tag{4-55}$$

上式中的积分可利用式(4-45)，即

$$\rho g h \iint_\Delta N_i \mathrm{d}x\mathrm{d}y = \rho g h \frac{1!0!0!}{(1+0+0+2)!} 2\Delta = \frac{1}{3}\rho g h \Delta \tag{4-56}$$

于是，均质等厚度平面三角形单元自重引起的等效结点力为

$$f_v^e = -\frac{1}{3}\rho g h \Delta [0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1] \tag{4-57}$$

从上式看，每个结点的等效结点力等于单元自重的三分之一。

下面是三角形单元自重的等效结点力程序：

```
1. function egf = EquivalentGravityForce( ie )
2. %   计算单元自重的等效结点力
3. %   输入参数
4. %      ie ------单元号
5. %   返回值
6. %      egf------自重的等效结点力
7.     global gElement gMaterial
8.     g = 9.8;
9.     area = ElementArea( ie ) ;
10.    h = gMaterial( gElement( ie, 4 ), 3 ) ;
11.    ro = gMaterial( gElement( ie, 4 ), 4 ) ;
12.    w = area * h * ro * g ;
13.    egf = -w/3 * [0; 1; 0; 1; 0; 1]
14. return
```

单元自重的等效结点公式(4-57)十分简单，因此程序亦如此。但是需要注意的一点是，工程中经常给出的是材料的比重（或容重），而不是密度，在准备数据时要进行换算。

4.5.2 线性分布表面力

设单元的 ij 边长度为 l，受到线性分布的表面力 $\boldsymbol{p} = [\sigma \quad \tau]^{\mathrm{T}}$，其中 σ 为法向压力，而 τ 为切向力，它们的正方向如图 4-7 所示。取局部坐标 s，其原点为

结点 i，由 i 指向 j 为正方向。则线性分布的压力表达式为

$$p = p_i + \frac{s}{l}(p_j - p_i) \tag{4-58}$$

所以分布力向量为

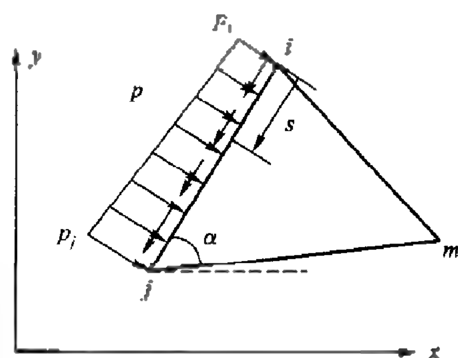


图 4-7　线性分布的压力

$$p = \begin{Bmatrix} p_x \\ p_y \end{Bmatrix} = \begin{Bmatrix} \sigma\sin\alpha - \tau\cos\alpha \\ -\sigma\cos\alpha - \tau\sin\alpha \end{Bmatrix} = \frac{1}{l}\begin{Bmatrix} \sigma(y_i - y_j) - \tau(x_i - x_j) \\ -\sigma(x_i - x_j) - \tau(y_i - y_j) \end{Bmatrix} \tag{4-59}$$

另外，根据面积坐标的定义，可以得到用局部坐标 s 表示的 3 个形函数在 ij 边上的值为

$$N_i = L_i = 1 - \frac{s}{l},\quad N_j = L_j = \frac{s}{l},\quad N_m = L_m = 0 \tag{4-60}$$

因此根据式(2-72)定义的表面力等效结点力公式，则有

$$f_{si} = \begin{Bmatrix} f_{pxi} \\ f_{pyi} \end{Bmatrix} = \int_0^l N_i \begin{Bmatrix} p_x \\ p_y \end{Bmatrix} h\mathrm{d}s \tag{4-61}$$

$$f_{sj} = \begin{Bmatrix} f_{pxj} \\ f_{pyj} \end{Bmatrix} = \int_0^l N_j \begin{Bmatrix} p_x \\ p_y \end{Bmatrix} h\mathrm{d}s \tag{4-62}$$

$$f_{sm}^e = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix} \tag{4-63}$$

因此，把式(4-58)代入式(4-61)和式(4-62)并完成积分运算，得到等效结点力为

$$\begin{aligned} f_s^e = &\frac{1}{6}h\left[(y_i - y_j)\sigma_1 \quad (x_j - x_i)\sigma_1 \quad (y_i - y_j)\sigma_2 \quad (x_j - x_i)\sigma_2 \quad 0 \quad 0\right]^{\mathrm{T}} \\ &+ \frac{1}{6}h\left[(x_j - x_i)\tau_1 \quad (y_j - y_i)\tau_1 \quad (x_j - x_i)\tau_2 \quad (y_j - y_i)\tau_2 \quad 0 \quad 0\right]^{\mathrm{T}} \end{aligned} \tag{4-64}$$

式中

$$\sigma_1 = 2\sigma_i + \sigma_j,\quad \sigma_2 = \sigma_i + 2\sigma_j,\quad \tau_1 = 2\tau_i + \tau_j,\quad \tau_2 = \tau_i + 2\tau_j \tag{4-65}$$

式中 σ_i、σ_j、τ_i 和 τ_j 分别是结点 i 和 j 的法向压力和切向力分量的大小。当表面力为均匀分布的压力时，即 $\sigma_i = \sigma_j = \sigma_0$，则有

$$f_s^e = \frac{1}{2}\sigma_0 h\left[(y_i - y_j) \quad (x_j - x_i) \quad (y_i - y_j) \quad (x_j - x_i) \quad 0 \quad 0\right]^{\mathrm{T}} \tag{4-66}$$

有时候,会遇到方向固定的表面力,比如设在单元的 ij 边界上作用沿 x 方向的线性分布荷载,如图 4-8 所示。在此情况下，表面荷载向量为

$$\boldsymbol{p} = \begin{Bmatrix} p_x \\ p_y \end{Bmatrix} = \begin{Bmatrix} p_i + (p_j - p_i)s/l \\ 0 \end{Bmatrix} \tag{4-67}$$

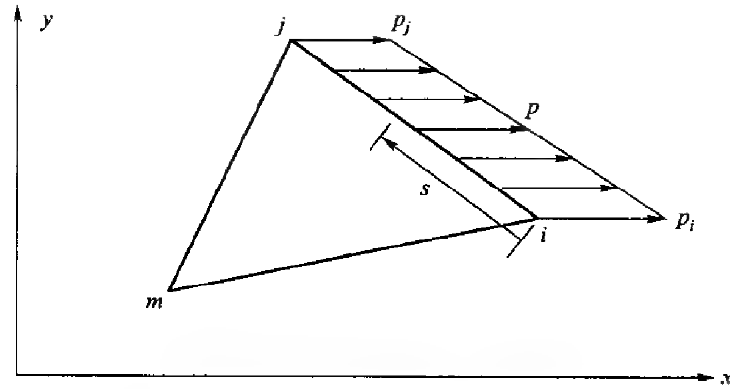


图 4-8 三角形分布荷载的等效结点力

形函数依然取式(4-60)，代入表面力的等效结点力公式(2-72)中得到

$$f_{pxi} = \int_0^l N_i p_x h\mathrm{d}s = \int_0^l \left(1 - \frac{s}{l}\right)\left[p_i + \frac{s}{l}(p_j - p_i)\right]h\mathrm{d}s = \frac{hl}{6}(2p_i + p_j) \tag{4-68}$$

$$f_{pxj} = \int_0^l N_j p_x h\mathrm{d}s = \int_0^l \frac{s}{l}\left[p_i + \frac{s}{l}(p_j - p_i)\right]h\mathrm{d}s = \frac{hl}{6}(p_i + 2p_j) \tag{4-69}$$

其余结点力分量都等于零。当为均匀分布荷载时，即当 $p_i = p_j = p_0$ 时，等效结点力为

$$\boldsymbol{f}_s = \frac{1}{2}phl\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{4-70}$$

即两个结点上的等效结点力是总荷载的一半。如果为三角形分布荷载，此时 $p_j = 0$，则有

$$\boldsymbol{f}_s = \frac{1}{2}phl\begin{bmatrix} \frac{2}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 \end{bmatrix} \tag{4-71}$$

从以上的几种荷载情况可以看出，对于常应变三角形单元按照等效结点力公式求得的结果，与按照静力平行分解方法求得的结果是一致的。

4.5.3 热应力

在第 3 章我们已经得到温度效应可以简化为由于温度引起的等效结点

力。对于平面问题，可以把式(3-126)简化为

$$f_T = \iint_{\Delta} B^{\mathrm{T}} D\alpha T h \mathrm{d}x\mathrm{d}y \tag{4-72}$$

对于各向同性材料

$$\alpha = \alpha \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^{\mathrm{T}} \tag{4-73}$$

把三角形单元的应变矩阵 B，弹性矩阵 D 代入即能得到温度荷载的等效结点力。例如对于平面应力问题，则有

$$f_T^e = \iint_{\Delta} B^{\mathrm{T}} D\alpha T \begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix} h \mathrm{d}x\mathrm{d}y = \frac{Eh\alpha}{2(1-\mu)\Delta} \begin{bmatrix} b_i & c_i & b_j & c_j & b_m & c_m \end{bmatrix}^{\mathrm{T}} \iint_{\Delta} T \mathrm{d}x\mathrm{d}y \tag{4-74}$$

如果温度 T 的分布函数为已知时，上式中的积分总可以用数值积分求得。特别当它是 x、y 的多项式时，则很容易写出精确积分的表达式。对于 T 为线性分布时，有

$$\iint_{\Delta} T \mathrm{d}x\mathrm{d}y = \frac{1}{3}(T_i + T_j + T_m)\Delta \tag{4-75}$$

式中，T_i、T_j 和 T_m 分别是结点 i、j 和 m 处的温度。在此情况下，热应力的等效结点力荷载列阵为

$$f_T^e = \frac{E\alpha h(T_i + T_j + T_m)}{6(1-\mu)} \begin{bmatrix} b_i & c_i & b_j & c_j & b_m & c_m \end{bmatrix}^{\mathrm{T}} \tag{4-76}$$

根据结点位移计算单元应力时，必须扣除温度的影响，即

$$\begin{aligned} \sigma &= D(\varepsilon - \varepsilon_0) = D\left(B\delta^e - \alpha T \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^{\mathrm{T}}\right) \\ &= DB\delta^e - \frac{E\alpha}{3(1-\mu)}(T_i + T_j + T_m)\begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^{\mathrm{T}} \end{aligned} \tag{4-77}$$

对于平面应变问题，只要在平面应力问题的公式中用 $E/(1-\mu^2)$ 代替 E，用 $\mu/(1-\mu)$ 代替 μ，以及 $(1+\mu)\alpha$ 代替 α 便可得到。经过这样的替换后，等效结点荷载公式为

$$f_T^e = \frac{E\alpha h(T_i + T_j + T_m)}{6(1-2\mu)} \begin{bmatrix} b_i & c_i & b_j & c_j & b_m & c_m \end{bmatrix}^{\mathrm{T}} \tag{4-78}$$

相应地，应力公式为

$$\sigma = DB\delta^e - \frac{E\alpha(T_i + T_j + T_m)}{3(1-2\mu)} \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}^{\mathrm{T}} \tag{4-79}$$

4.5.4 初应力

前面的温度荷载可以看成是初应变引起的等效结点力。当结构存在初应力时，同样可以通过等效结点力的方法来处理。此时，应力应变关系为

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\varepsilon} + \boldsymbol{\sigma}_0 \tag{4-80}$$

式中 $\boldsymbol{\sigma}_0$ 即为结构加载前就存在的初应力。用上式来代替第 2 章中用最小势能原理推导有限元基本方程中的应力应变关系式（2-48），将会得到初应力的等效结点力为

$$\boldsymbol{f}_{\sigma}^{e} = -\iiint_{\Omega} \boldsymbol{B}^{\mathrm{T}} \boldsymbol{\sigma}_0 \,\mathrm{d}\Omega \tag{4-81}$$

对于三角形单元，如果假定初应力场在三角形单元内是线性分布的，则

$$\boldsymbol{f}_{\sigma}^{e} = -\frac{\Delta h}{3}\boldsymbol{B}^{\mathrm{T}}(\boldsymbol{\sigma}_i + \boldsymbol{\sigma}_j + \boldsymbol{\sigma}_m) \tag{4-82}$$

式中 $\boldsymbol{\sigma}_i$、$\boldsymbol{\sigma}_j$ 和 $\boldsymbol{\sigma}_m$ 分别是三角形单元结点 i、j 和 m 上的初应力向量，$\boldsymbol{B}$ 是三角形单元的应变矩阵，由式（4-18）定义，Δ 和 h 分别是三角形单元的面积和厚度。

跟初应变效应类似，最后计算应力时，要用式（4-80）来代替（4-21）。

4.6 矩形单元

由于前面所述的三角形单元是常应变常应力单元，其精度受到一定的限制。采用更高阶数位移模式的矩形单元，可以更好地反映弹性体中的位移状态和应力状态。但是它对于几何形状复杂的模型难于采用，因此经常把它和三角形单元组合起来使用，互补不足，如图 4-9 所示。

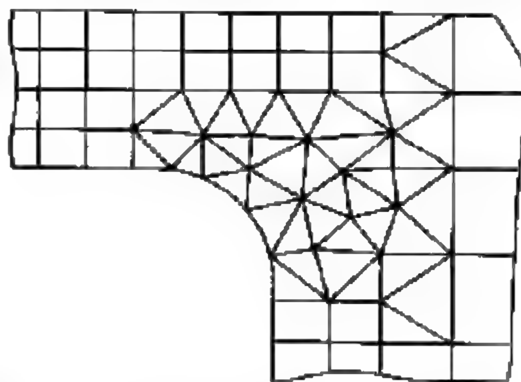


图 4-9 混合使用矩形单元和三角形单元

4.6.1 位移模式

设有图 4-10 中所示矩形单元 1234，其边长分别是 $2a$、$2b$，矩形的两边分别与 x、y 轴平行。取矩形的 4 个角点为结点，每个结点有两个位移分量，整个矩形单元共有 8 个自由度。我们可以用分析三角形单元相似的方法来分析该单元的力学特性。但是，如果引入一个局部

坐标系 $\xi\eta$，则可以得到较为简洁的结果。

如图 4-10 所示，局部坐标的原点取在矩形的形心，ξ 和 η 轴分别与整体坐

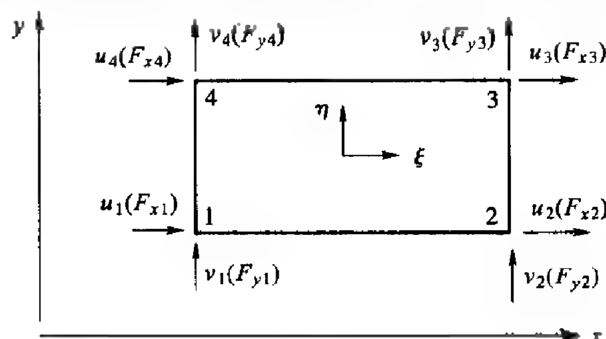


图 4-10 矩形单元

标轴 x 和 y 平行，其坐标变换关系为

$$x = x_0 + a\xi \qquad y = y_0 + b\eta \tag{4-83}$$

式中

$$x_0 = (x_1 + x_2)/2 = (x_3 + x_4)/2 \tag{4-84}$$

$$y_0 = (y_1 + y_2)/2 = (y_3 + y_4)/2 \tag{4-85}$$

$$a = (x_2 - x_1)/2 = (x_4 - x_3)/2 \tag{4-86}$$

$$b = (y_2 - y_1)/2 = (y_4 - y_3)/2 \tag{4-87}$$

其中$(x_i, y_i)(i=1,2,3,4)$是结点 i 的整体坐标。

在局部坐标系中，结点 i 的坐标是$(\xi_i, \eta_i)(i=1,2,3,4)$，它们的值分别是 ± 1。例如 $\xi_1 = -1$，$\eta_1 = -1$。取位移模式

$$\begin{aligned} u &= \alpha_1 + \alpha_2\xi + \alpha_3\eta + \alpha_4\xi\eta \\ v &= \alpha_5 + \alpha_6\xi + \alpha_7\eta + \alpha_8\xi\eta \end{aligned} \tag{4-88}$$

将结点的局部坐标值和 4 个结点处位移分量的值代入上式，则可以获得两组四元联立方程，从而可以分别解出未知参数 $\alpha_1, \alpha_2, \cdots, \alpha_8$，便可得到用结点位移表示的位移模式

$$u = \sum_{i=1}^{4} N_i u_i, \quad v = \sum_{i=1}^{4} N_i v_i \tag{4-89}$$

式中 $$N_i = (1 + \xi_i\xi)(1 + \eta_i\eta)/4 \qquad (i = 1,2,3,4) \tag{4-90}$$

或者写成与前面一样的形式

$$
\boldsymbol{u} = \begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} \boldsymbol{N}_1 & \boldsymbol{N}_2 & \boldsymbol{N}_3 & \boldsymbol{N}_4 \end{bmatrix} \begin{Bmatrix} \boldsymbol{\delta}_1 \\ \boldsymbol{\delta}_2 \\ \boldsymbol{\delta}_3 \\ \boldsymbol{\delta}_4 \end{Bmatrix} = \sum_{i=1}^{4} \boldsymbol{N}_i \boldsymbol{\delta}_i \tag{4-91}
$$

式中

$$
\boldsymbol{N}_i = \begin{bmatrix} N_i & 0 \\ 0 & N_i \end{bmatrix} = N_i \boldsymbol{I}, \quad \boldsymbol{\delta}_i = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix} \quad (i = 1,2,3,4) \tag{4-92}
$$

4.6.2 应变、应力和单元刚度矩阵

利用几何方程求出单元的应变

$$
\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \partial u / \partial x \\ \partial v / \partial y \\ \partial u / \partial y + \partial v / \partial x \end{Bmatrix} = \frac{1}{ab} \begin{Bmatrix} b \partial u / \partial \xi \\ a \partial v / \partial \eta \\ a \partial u / \partial \eta + b \partial v / \partial \xi \end{Bmatrix} \tag{4-93}
$$

将式(4-89)代入上式,得到

$$
\boldsymbol{\varepsilon} = [\boldsymbol{B}_1 \quad \boldsymbol{B}_2 \quad \boldsymbol{B}_3 \quad \boldsymbol{B}_4] \boldsymbol{\delta}^e \tag{4-94}
$$

式中

$$
\boldsymbol{B}_i = \frac{1}{ab} \begin{bmatrix} b \dfrac{\partial N_i}{\partial \xi} & 0 \\ 0 & a \dfrac{\partial N_i}{\partial \eta} \\ a \dfrac{\partial N_i}{\partial \eta} & b \dfrac{\partial N_i}{\partial \xi} \end{bmatrix} = \frac{1}{4ab} \begin{bmatrix} b \xi_i (1 + \eta_i \eta) & 0 \\ 0 & a \eta_i (1 + \xi_i \xi) \\ a \eta_i (1 + \xi_i \xi) & b \xi_i (1 + \eta_i \eta) \end{bmatrix} \quad (i = 1,2,3,4) \tag{4-95}
$$

利用应力应变关系，可以得到用结点位移表示的单元应力

$$
\boldsymbol{\sigma} = \boldsymbol{D} \boldsymbol{\varepsilon} = \begin{bmatrix} \boldsymbol{S}_1 & \boldsymbol{S}_2 & \boldsymbol{S}_3 & \boldsymbol{S}_4 \end{bmatrix} \boldsymbol{\delta}^e \tag{4-96}
$$

式中

$$
\boldsymbol{S}_i = \boldsymbol{D} \boldsymbol{B}_i \quad (i = 1,2,3,4) \tag{4-97}
$$

对于平面应力问题

$$
\boldsymbol{S}_i = \frac{E}{1 - \mu^2} \frac{1}{4ab} \begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & \dfrac{1 - \mu}{2} \end{bmatrix} \begin{bmatrix} b \xi_i (1 + \eta_i \eta) & 0 \\ 0 & a \eta_i (1 + \xi_i \xi) \\ a \eta_i (1 + \xi_i \xi) & b \xi_i (1 + \eta_i \eta) \end{bmatrix}
$$

$$
= \frac{E}{4ab(1-\mu^2)} \begin{bmatrix} b\xi_i(1+\eta_i\eta) & a\mu\eta_i(1+\xi_i\xi) \\ \mu b\xi_i(1+\eta_i\eta) & a\eta_i(1+\xi_i\xi) \\ \frac{1-\mu}{2}a\eta_i(1+\xi_i\xi) & \frac{1-\mu}{2}b\xi_i(1+\eta_i\eta) \end{bmatrix} \tag{4-98}
$$

若将单元刚度矩阵写成如下的分块形式

$$
\boldsymbol{K}^e = \begin{bmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} & \boldsymbol{K}_{13} & \boldsymbol{K}_{14} \\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22} & \boldsymbol{K}_{23} & \boldsymbol{K}_{24} \\ \boldsymbol{K}_{31} & \boldsymbol{K}_{32} & \boldsymbol{K}_{33} & \boldsymbol{K}_{34} \\ \boldsymbol{K}_{41} & \boldsymbol{K}_{42} & \boldsymbol{K}_{43} & \boldsymbol{K}_{44} \end{bmatrix} \tag{4-99}
$$

则其中的子矩阵可由下式计算

$$
\boldsymbol{K}_{ij} = \iint \boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}_j\, h\mathrm{d}x\mathrm{d}y = ab\int_{-1}^{1}\int_{-1}^{1} \boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}_j\, h\mathrm{d}x\mathrm{d}y \qquad (i,j = 1,2,3,4) \tag{4-100}
$$

若单元厚度 h 为常数，则可以得到单元刚度矩阵的显式如下：

$$
\boldsymbol{K}_{ij} = \frac{Eh}{4(1-\mu^2)}
$$

$$
\begin{bmatrix} \frac{b}{a}\xi_i\xi_j(1+\frac{1}{3}\eta_i\eta_j) + \frac{1-\mu}{2}\frac{a}{b}\eta_i\eta_j(1+\frac{1}{3}\xi_i\xi_j) & \mu\xi_i\eta_j + \frac{1-\mu}{2}\eta_i\xi_j \\ \mu\eta_i\xi_j + \frac{1-\mu}{2}\xi_i\eta_j & \frac{a}{b}\eta_i\eta_j(1+\frac{1}{3}\xi_i\xi_j) + \frac{1-\mu}{2}\frac{b}{a}\xi_i\xi_j(1+\frac{1}{3}\eta_i\eta_j) \end{bmatrix}
$$

$$
(i,j=1,2,3,4) \tag{4-101}
$$

对于平面应变问题的公式，只要把上式中的 E 换成 $E/(1-\mu^2)$，μ 换成 $\mu/(1-\mu)$ 即可。

平面应力矩形单元的单元刚度子矩阵的推导程序如下：

```
syms a b xi ei xj ej x eta E u h
Bi = 1/4/a/b* [b* xi* (1 + ei* eta)          0
                        0                a* ei* (1 + xi* x)
               a* ei* (1 + xi* x)       b* xi* (1 + ei* eta) ];
Bj = 1/4/a/b* [b* xj* (1 + ej* eta)          0
                        0                a* ej* (1 + xj* x)
               a* ej* (1 + xj* x)       b* xj* (1 + ej* eta) ];
D = E/(1 - u^2)* [1      u      0
                  u      1      0
```

```
          0     0   (1 - u)/2];
krs = a* b* h* int(int(transpose(Bi)* D* Bj, x, -1, 1), eta, -1, 1);
[illegible]
```

上面的各变量 a,b,xi,ei,xj,ej,x,eta,E,u 和 h 分别是单元尺寸参数 a,b,第 i,j 节点的局部坐标 ξ_i,η_i,ξ_j,η_j,局部坐标 ξ,η 和材料弹性模量 E,泊松比 μ 和单元厚度 h。MATLAB 给出的单元刚度子矩阵的元素是

```
>> krs(1,1)
ans =
1/24/b/a* h* E* (-2* xi* ei* xj* ej* b^2 - ei* xi* ej* [illegible]
    ej* xj* a^2* u [illegible] b^2 - 3* [illegible])/
    (-1 + u^2)
>> krs(1,2)
ans =
[illegible] h* E* (2* xi* u* ej + ei* xj - ei* xj* u)/(-1 + u^2)
>> krs(2,1)
ans =
[illegible]
>> krs(2,2)
ans =
-1/24/b/a* h* E* (xi* ei* xj* ej* b^2 - xi* ei* xj* ej* b^2* u + 2* ei*
    xi* ej* xj* a^2 + 6* ei* ej* a^2 + 3 [illegible]
    (-1 + u^2)
```

通过比较发现 MATLAB 的计算结果跟公式(4-101)给出的是一样的。这里需要注意的是,MATLAB 的简化函数 simple 不是万能的,它不能得到像式(4-101)一样简洁的结果。这也是目前具有符号运算功能的软件的不足之处,希望能在不久的将来解决这个问题。

从前面的推导过程可以看出,矩形单元的位移模式(4-88)比三角形单元中采用的线性位移模式增加了 $\xi\eta$ 项(相当于 xy 项),这种位移模式可以称为双线性模式。可以矩形单元内的应变分量不是常量,这点也可以由应变矩阵 $\boldsymbol{B}$ 和应力矩阵 $\boldsymbol{S}$ 的表达式中的元素含有变量 ξ 和 η 可知。因此一般情况下,在弹性体中采用相同数目的结点时,矩形单元的精度要比常应变三角形单元高。另外,位移模式(4-88)中的 α_1,α_2,α_3,α_5,α_6 和 α_7 与三角形单元相同,它们反映了刚体位移和常应变。而且在单元的边界($\xi = \pm 1$ 或者 $\eta = \pm 1$)上,位移是按线性变化的,可见两个相邻单元的位移在公共边界上是连续的。

4.6.3 等效结点力

矩形单元的荷载列阵 f^e 的计算与三角形单元的类似,可利用普遍公式(2-72)计算分布面力和体积力的等效结点力，温度荷载和初应力荷载则为式(3-126)和(4-81),然后叠加得到。因为矩形单元有 4 个结点,因此 f^e 有 8 个元素,即

$$f^e = \begin{bmatrix} F_{x1} & F_{y1} & F_{x2} & F_{y2} & F_{x3} & F_{y3} & F_{x4} & F_{y4} \end{bmatrix}^T \tag{4-102}$$

现给出几种常见荷载的结果。

1. 单元的自重

根据式(2-72),我们计算每一个结点 y 方向的等效结点力为

$$F_{yi} = -\rho gh \iint_{\Omega} N_i \mathrm{d}x\mathrm{d}y = -\rho gabh \int_{-1}^{1}\int_{-1}^{1} N_i \,\mathrm{d}\xi \mathrm{d}\eta \qquad (i = 1,2,3,4) \tag{4-103}$$

把式(4-90)代入上式,完成积分得到

$$F_{yi} = -\rho gabh = -W/4 \qquad (i = 1,2,3,4) \tag{4-104}$$

式中 $W = 4\rho gabh$ 为单元的总重量。而 x 方向的结点力为零，因此单元自重的等效结点力为

$$f_v = -W\begin{bmatrix} 0 & 1/4 & 0 & 1/4 & 0 & 1/4 & 0 & 1/4 \end{bmatrix}^T \tag{4-105}$$

即把单元的自重平均分配给 4 个结点。

2. 线性分布的表面力

如果单元的一个边界受到线性分布的表面力，如图 4-11 所示,不妨设荷载作用在 23 边上,结点 2 和 3 上的分布力大小分别为 p_2 和 p_3,那么就可以把

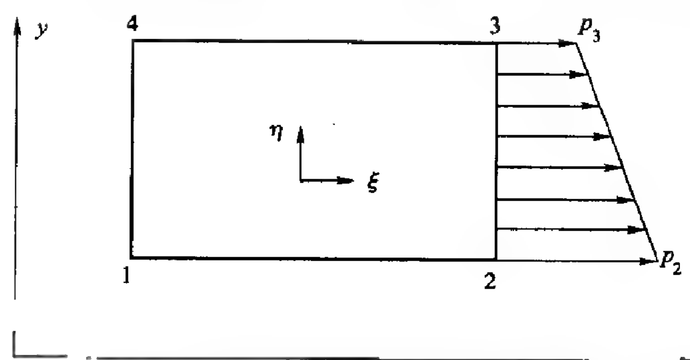


图 4-11　受线性分布力作用的矩形单元

分布力用局部坐标表示为

$$p = p_2 + \frac{1}{2}(\eta + 1)(p_3 - p_2) \tag{4-106}$$

根据式(2-72),在结点 2 上 x 方向上的等效结点力为

$$F_{x2} = h\int_l N_2 p\mathrm{d}y = hb\int_{-1}^{1} \frac{1}{2}(1+\eta)\left[p_2 + \frac{1}{2}(\eta+1)(p_3 - p_2)\right]\mathrm{d}\eta = \frac{1}{3}hb(2p_2 + p_3) \tag{4-107}$$

同样可以得到结点 3 上 x 方向上的等效结点力为

$$F_{x3} = \frac{1}{3}hb(p_2 + 2p_3) \tag{4-108}$$

当分布力为三角形分布时,即 $p_3 = 0$,那么在结点 2 上的等效结点力是总荷载的 2/3,在结点 3 上的等效结点力是总荷载的 1/3

3. 温度荷载

温度荷载的等效结点力公式为式(3-126),与三角形单元类似,必须要给出单元内的温度分布模式。为了分析方便,可以假定单元内温度跟位移有同样的分布形式,这样就能直接利用位移插值的形函数,即

$$T = \sum_{j=1}^{4} N_j T_j \tag{4-109}$$

从而得到温度荷载的等效结点力为

$$\boldsymbol{f}_{T_i} = \begin{Bmatrix} F_{xi} \\ F_{yi} \end{Bmatrix} = abh\int_{-1}^{1}\int_{-1}^{1} \boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{D}\begin{Bmatrix} 1 \\ 1 \\ 0 \end{Bmatrix} T\mathrm{d}\xi\mathrm{d}\eta \qquad (i = 1,2,3,4) \tag{4-110}$$

把式(4-90)、式(4-95)和式(4-109)代入上式,并完成积分,得到

$$\boldsymbol{f}_{T_i} = \frac{E\alpha h}{4(1-\mu)}\begin{Bmatrix} b\sum_{j=1}^{4} \xi_i\left(1 + \frac{1}{3}\eta_i\,\eta_j\right)T_j \\ a\sum_{j=1}^{4} \eta_i\left(1 + \frac{1}{3}\xi_i\,\xi_j\right)T_j \end{Bmatrix} \qquad (i = 1,2,3,4) \tag{4-111}$$

上式就是矩形单元温度荷载的等效结点力。

4.7 其他注意事项

4.7.1 收敛准则

对于一个数值方法,我们总是希望随着网格的逐步细分,得到的解答收敛于问题的精确解。从上面分析中可以看出,在单元形状确定以后,位移模式的

选择是关键。等效荷载的计算、刚度矩阵的建立和应变应力的计算等，都依赖于位移模式。显然，一个与真实位移分布有很大差别的位移模式，是很难得到很好的数值结果。为了保证解答的收敛性，要求位移模式必须满足以下 3 个条件：

(1) 位移模式必须包含单元的刚体位移：当结点位移是由某个刚体位移所引起时，弹性体内不会有应变。这样，位移模式就不但要具有描述单元本身变形的能力，而且还要具有描述由于其他单元变形通过结点位移传递引起单元刚体位移的能力。

(2) 位移模式必须包含单元的常应变：每个单元的应变一般总是包含两个部分：一部分是与该单元中各点的位置坐标有关的(即所谓各点的变应变)；另一部分是与位置无关的（即所谓常应变）。从物理意义上看，当单元尺寸无限缩小时，每个单元中的应变应趋于常量。因此单元的位移模式必须包含这些常应变，否则就没有可能收敛于精确解。不难看出，前面讨论的几种单元都符合这些条件。

(3) 位移模式在单元内要连续，相邻单元间要协调：当选择多项式来构成位移模式时，单元内的连续性要求总是满足的。单元间的协调性要求单元之间不开裂也不重叠。通常，当单元交界面上的位移取决于该交界面上结点的位移时，可以保证位移的协调性。

在有限单元法中，满足条件 1 和 2 的单元，称为完备单元；满足条件 3 的单元，称为协调单元或保续单元。前面已经讨论过的杆系单元、三角形单元和矩形单元，显然同时满足 3 个条件，因此是完备的协调单元。满足条件 1 和 2 而不能满足条件 3 的单元，称为完备非协调单元。这种单元的缺点是不能保证计算结果的收敛性，但是，有时这类单元的收敛速度反而比协调单元更快。

事实上已经证明，对于完备的协调单元，其刚度系数的数值比精确的要大。这样一来，在给定的荷载之下，计算模型的变形比实际结构的要小。因此，当单元网格分割得越来越细时，位移的近似解将由下方收敛于精确解。

4.7.2　多项式位移模式阶次的选择

前面我们讨论了选择多项式位移模式的阶次时，要考虑到完备性和协调性的要求。实践证明，除了这两项以外，还要考虑位移模式应该与局部坐标系的方位无关，这一性质称为几何各向同性。对于线性多项式，各向同性的要求就等价于必须包含常应变状态。对于高次模式，就是不应有一个偏惠的坐标

方向，也就是位移形式不应随局部坐标的变换而改变。经验证实，实现几何各向同性的一种方法，是根据巴斯卡三角形（图 4-12）来选择完全二维多项式的

对称轴

							名称	项数
			1				常数项	1
		x		y			线性项	3
		x^2	xy	y^2			二次项	6
	x^3	x^2y		xy^2	y^3		三次项	10
	x^4	x^3y	x^2y^2	xy^3	y^4		四次项	15
x^5	x^4y	x^3y^2		x^2y^3	xy^4	y^5	五次项	21

图 4-12 巴斯卡三角形

各项。在二维多项式中，若包含三角形对称轴一边的任意一项则必须同时包含另一边的对称项。

4.7.3 对称性的利用

很多实际结构在空间上是对称的，如果所考虑的荷载也具有对称或反对称的特性，那么就可以取该结构的一部分为计算模型，从而大大减少计算和前后处理的工作量。图 4-13 所示为受纯弯曲的梁，它对于 x 和 y 轴都对称，而荷载对于 y 轴对称，x 轴反对称。因此，变形和应力也具有同样的对称和反对称性质，所以只需计算梁的四分之一就可以了。对于删去部分结构的影响用剩下部分的边界条件来考虑。

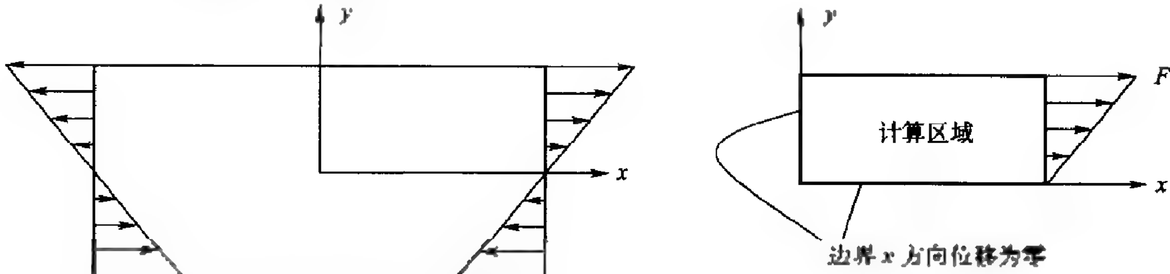


图 4-13 对称性的利用

4.7.4 应力计算结果的整理

应力计算结果主要包括位移和应力两个方面。在位移方面一般无需进行整理工作。但是由于应力在单元间不连续，一般要做一定的修正。比如，前面

提到的三角形常应变单元也是常应力单元。算出的应力，通常都可作为单元形心处的应力。为了由计算结果得到弹性体内某一点的接近实际的应力值，通常可采用绕结点平均法或两单元平均法。

所谓绕结点平均法，就是把环绕某一结点的各单元常应力加以平均，用以表示该结点的应力。为了使这样的平均得到的应力能够较好地表示结点处的实际应力，环绕该结点的各个单元的面积不应相差太大。单元之间的尺寸相差较多时，采用单元面积作为权系数的加权平均更为合理。

绕结点平均法计算出来的结果，在内结点处较好，而在边界结点处可能很差。因此，边界结点处的应力不宜直接由单元应力平均来获得，而应由内结点的应力外推算出来。

下面是采用绕结点加权平均法计算三角形单元结点应力的程序段：

```
1.gNodeStress = zeros(node_number, 6) ;
2.for i=1:node_number
3.    S = zeros(1,3) ;
4.    A = 0 ;
5.    for ie=1:element_number
6.        for k=1:3
7.            if i == gElement(ie,k)
8.                area = ElementArea(ie) ;
9.                S = S + gElementStress(ie,1:3) * area ;
10.               A = A + area ;
11.               break ;
12.           end
13.       end
14.   end
15.   gNodeStress(i,1:3) = S / A ;
16.   gNodeStress(i,6) = 0.5*sqrt((gNodeStress(i,1)-gNodeStress(i,2))^2 + ...
17.                           4* gNodeStress(i,3)^2) ;
18.   gNodeStress(i,4)=0.5*(gNodeStress(i,1)+gNodeStress(i,2))+gNodeStress(i,6);
19.   gNodeStress(i,5)=0.5*(gNodeStress(i,1)+gNodeStress(i,2))-gNodeStress(i,6);
20.end
```

第1行定义了结点应力矩阵，每一行保存每个结点的3个应力分量、2个主应力分量、1个最大剪应力分量，共6个。其中的 gElementStress 是单元应力矩阵，每一行表示一个单元的应力，所保存的应力分量与结点应力一样。第2~20行为对每一个结点循环，计

算包含该结点的所有单元的应力和面积，进行加权平均。其中第 3 和第 4 行对面积 A 和应力加权和对 S 赋初值，这里特别要指出的是，因为初学者经常在编制求和的程序时忘记对变量赋初值，导致错误结果。第 5 ~ 14 行是在所有单元中查找包含该结点的单元。第 15 行计算加权平均值。第 16 ~ 19 行根据平面应力状态计算主应力和最大剪应力。如果是平面应变状态，还存在第 3 主应力 σ_3，$\sigma_3 = -\mu(\sigma_1 + \sigma_2)$。

4.8 算例

4.8.1 受纯弯曲的梁

设有一承受纯弯曲的梁，其尺寸及荷载如图 4-14 所示，梁的厚度 $h = 1\text{cm}$，弹性模量 $E = 210\text{GPa}$，泊松比 $\mu = 0.3$。由于对称性和反对称性，仅需对梁的 1/4 进行计算。根据对称和反对称性，在 y 轴和 x 轴上，x 方向的位移均为零。另外，为了消除结构在 y 方向上的刚体位移，可令原点处 y 方向的位移为零。

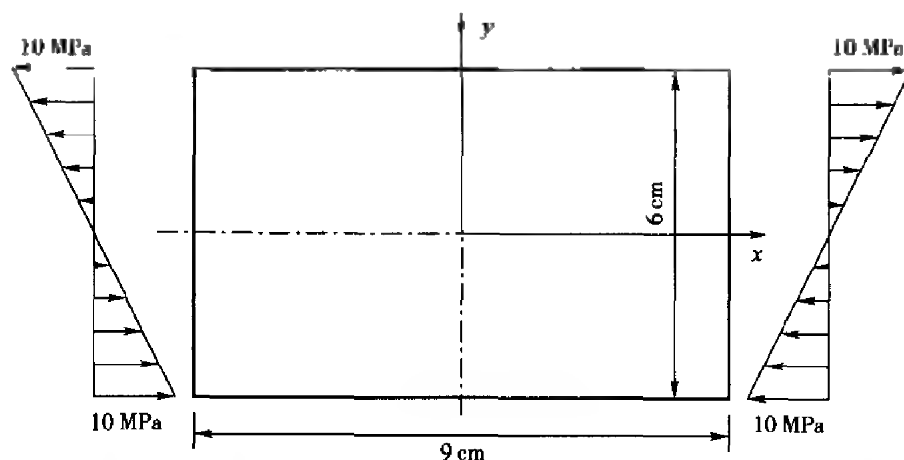


图 4-14 受纯弯曲的梁

由于结构十分规则，形状是矩形，因此可以采用矩形单元，在网格划分时采用自动划分的方法，只要指定水平方向和竖直方向的单元数目，结点坐标和单元定义都可自动进行，这样有利于考察不同单元数目对计算结果的影响。图 4-15 是计算网格分别为 $2\times2, 3\times3, \cdots, 16\times16$ 时 A 点 x 方向位移的计算精度，由图可以发现有限元结果总是小于精确解[2]，但是随网格的加密，单调收敛于精确解。另外，当单元数达到一定数量后，再加密网格对提高计算精度效果已不显著。图 4-16 给出了单元形状对计算精度的影响，随着长宽比的增

加，A点x方向位移的计算精度逐渐下降，虽然单元数量基本保持不变。因此在建立有限元模型时，网格的密度和单元的形状要同时兼顾，以保证计算精度。

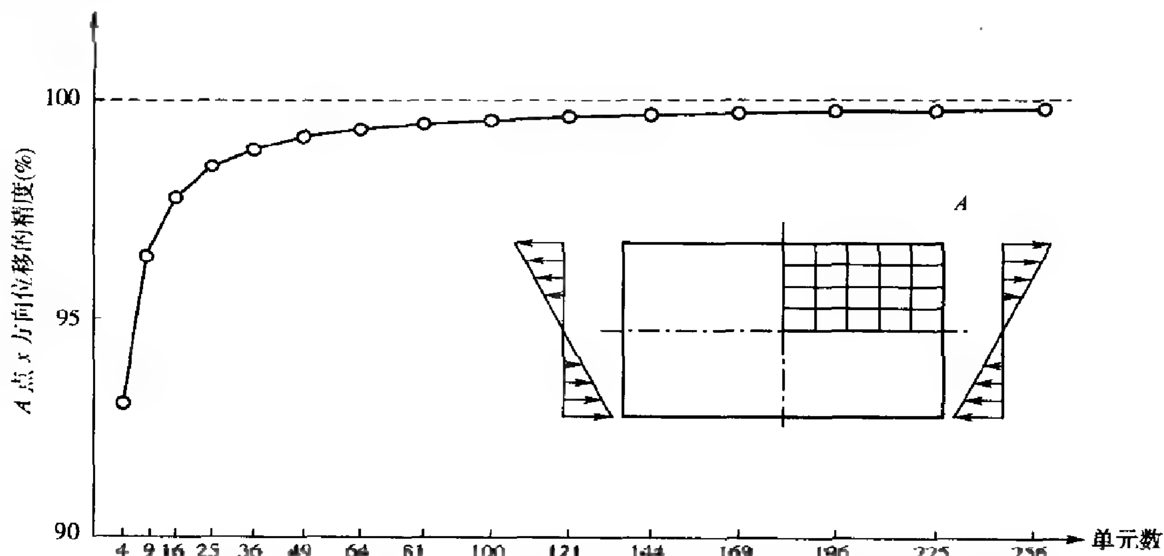


图4-15 结点数与计算精度的关系

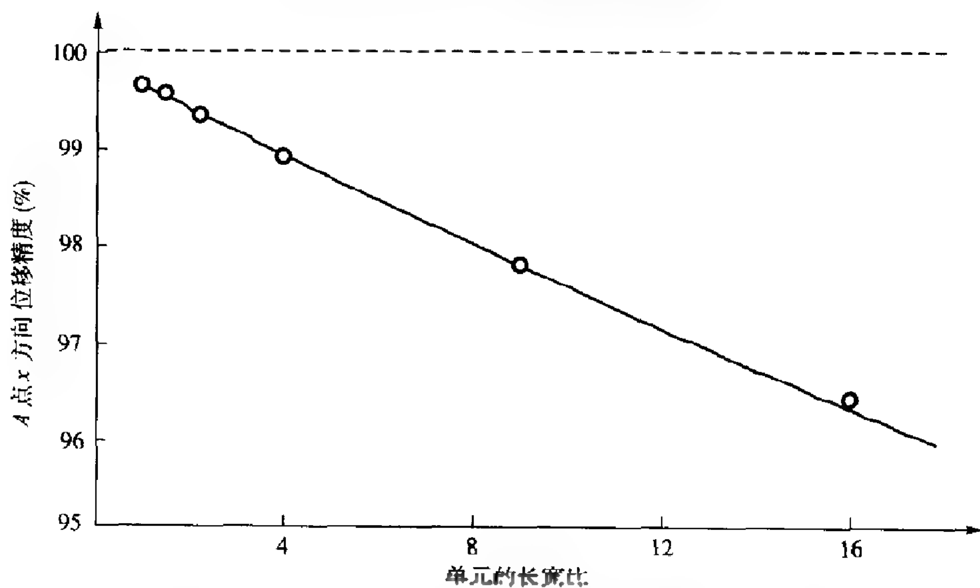


图4-16 单元的长宽比对有限元计算结果精度的影响

4.8.2 在自重作用下隧道的围岩和衬砌的变形

设有一双连拱形隧道，其基本构造如图4-17所示。现利用平面应变假定分析其在围岩和衬砌自重作用下的变形和应力分布，分析采用三角形平面单元。由于主要关心衬砌附近的变形和应力，因此采用有梯度的网格布局，在衬砌附

近的网格密度要远远大于围岩部分的网格。这样既可以保证区域的计算精度，又可以大大节省数据处理工作和计算时间。有限元网格如图 **4-18** 和图

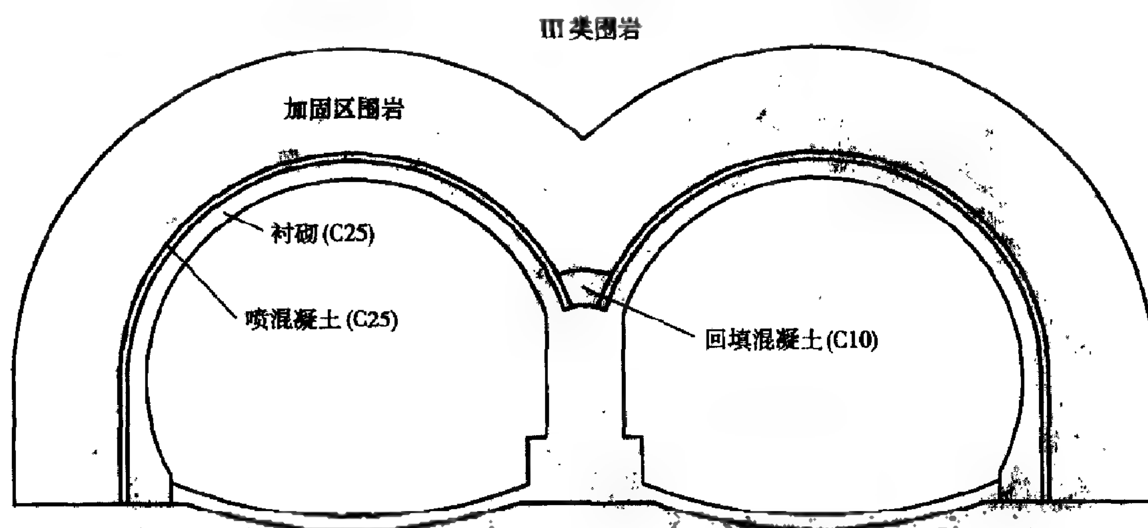


图 4-17 双连拱隧道

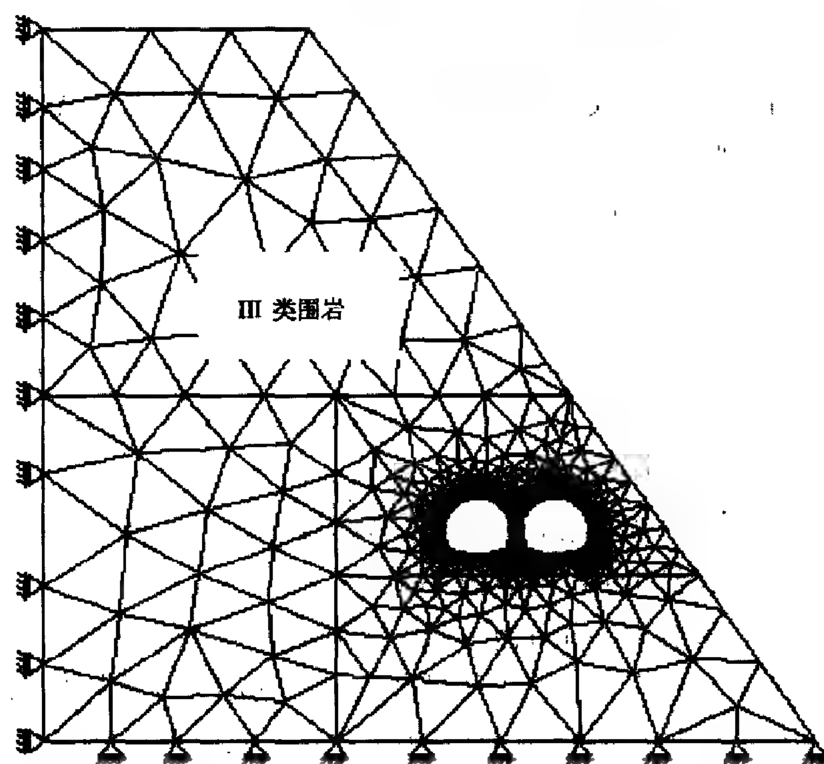


图 4-18 隧道的三角形平面网格模型和边界条件

4-19所示，共 **2729** 个结点，**5145** 个单元。由于结点数和单元数很多，以文本形式在屏幕上或文件中显示应力值和位移值很不直观，而采用应力云图或位移云图或等值线的图形方式则更为有效。幸运的是，**MATLAB** 具有强大的图形

编程能力，因此通过简单的语句就可以轻而易举地完成用其他语言难于完成的工作。

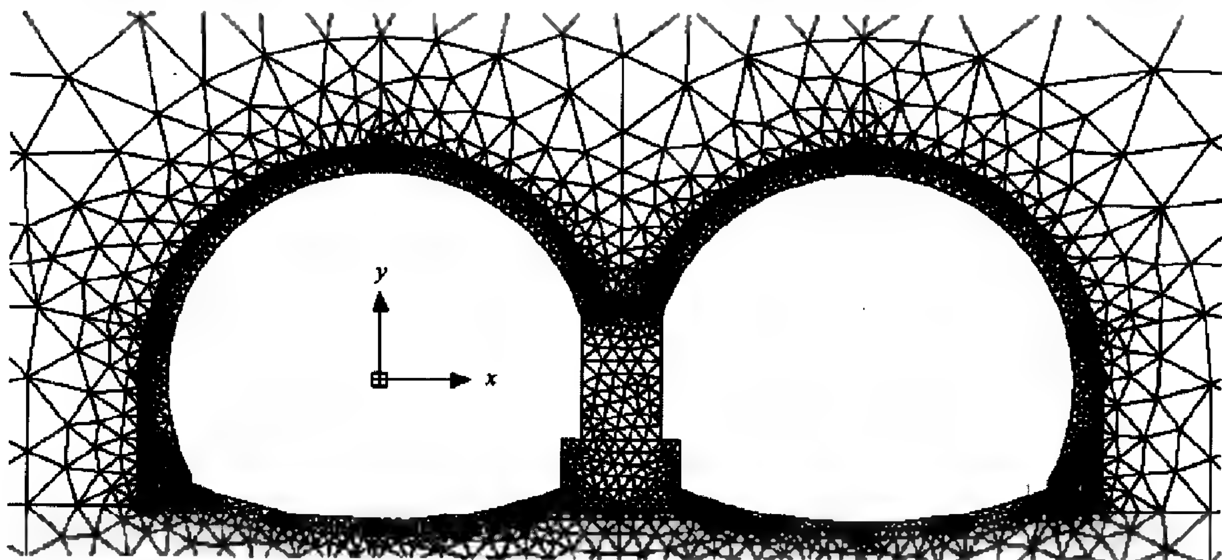


图 4-19 衬砌附近的网格示意图

这里主要介绍 **MATLAB** 中两个主要的绘图函数 **line** 和 **patch**。从函数名字就应该能猜出 **line** 是画线函数，而 **patch** 则是绘制多边形函数。这里只对它们作简单的介绍，详细的说明请查阅 **MATLAB** 的"帮助"文档。下面用具体的实例来说明 **line** 的用法。例如用 **line** 命令在 **MATLAB** 的图形窗口中绘制一个表示 x 和 y 方向位移被固定的标志，如图 **4-20** 所示。

line 函数的参数可以是两个向量，它们分别是对应端点的 x 轴和 y 轴的坐标，根据这些坐标系统就绘制折线段。它们也可以是两个维数相等的矩阵，每一列是一个折线段。如在图 **4-20** 中的 **6** 条斜线就可以用两个 **2** 行 **6** 列的矩阵 **X** 和 **Y** 来保存它们端点坐标，然后用一条 **line** 命令一次全部绘制出来。另外，它们还可以指定线条的颜色、线形、粗细等属性，这些在 **MATLAB** 软件附带的"帮助"文档中均有详细介绍，这里就不赘述。

另一个有用的函数是 **patch**，它可以绘制用一定颜色填充的多边形，内部既可以用均匀的颜色填充，也可以根据多边形顶点颜色利用内插方法，用渐变的颜色填充。因此它非常适合用于绘制应力云图或位移云图。例如采用以下几行简单的语句。

```
colormap('default');
x=[0, 1, 0.5];
y=[0, 0, 0.866];
c=[-5; 5; 0];
```

```
patch( x, y, c) ;
```

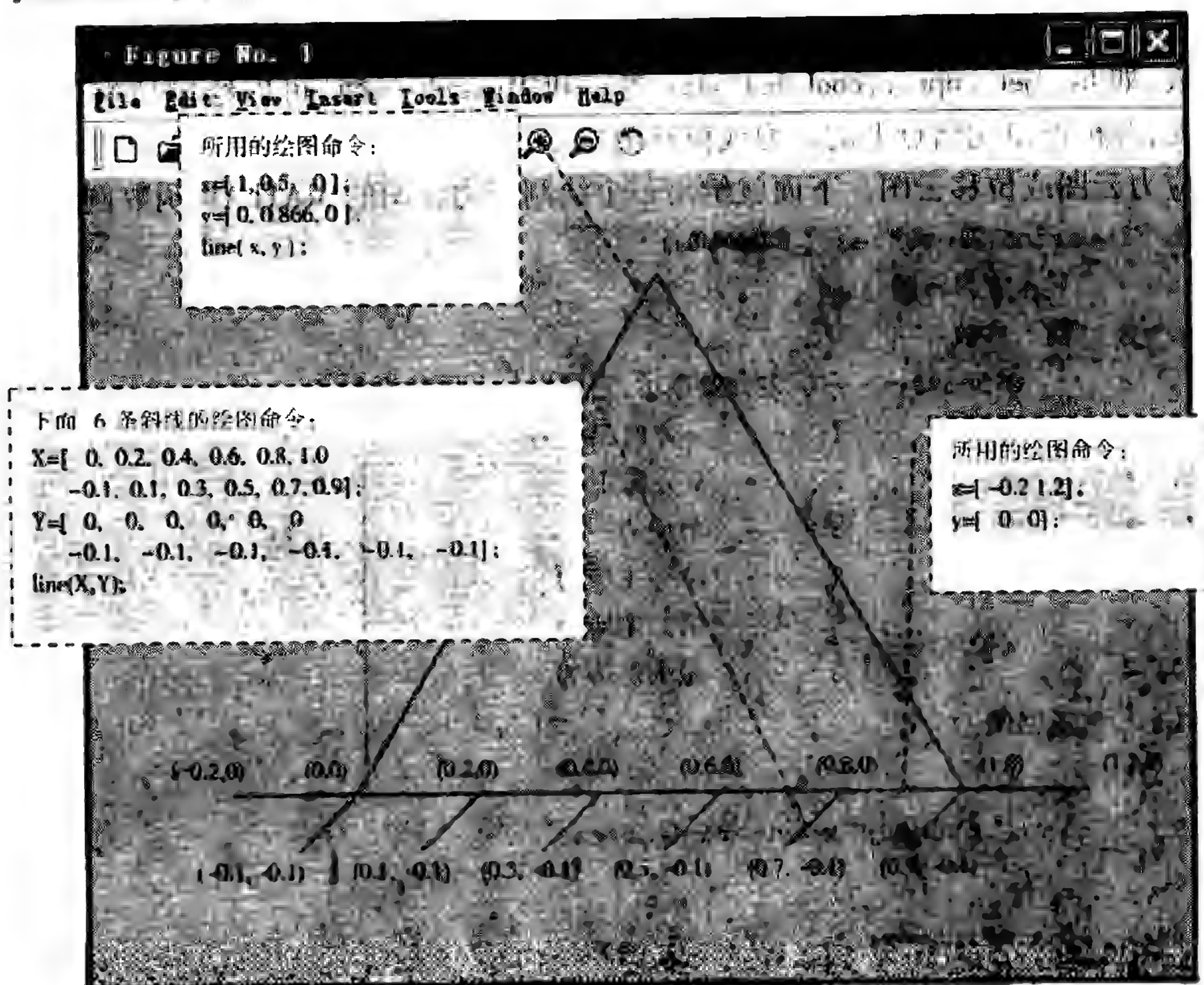


图 4-20 MATLAB 的 line 函数用法

就能得到如图 4-21 所示的用颜色填充的三角形。patch 函数的第 3 个参

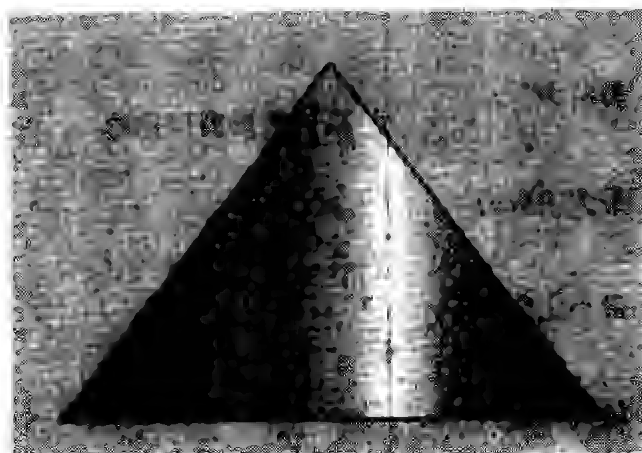


图 4-21 用 patch 函数生成的填充三角形

数 c给定了 3 个角点的颜色值。内部填充的颜色由内插得到，颜色值所对应的颜色由第 i 行的 colormap指定。MATLAB提供了预先定义的多种颜色映射表，如 hsv、jet、copper、cool、hot、gray 等。利用 patch 函数，可以绘制所有的单元，并根据单元结点上的应力或位移值指定单元结点上的颜色值，从而可得到应力云图或位移云图。下面就给出一个绘制应力云图的 MATLAB 函数例子。

```
1.  function PlotStress( iStress )
2.  %   显示应力云图
3.  %   输入参数：
4.  %       iStress ---应力分量指示，它可以是下面的值
5.  %                   1  --  x方向正应力
6.  %                   2  --  y方向正应力
7.  %                   3  --  剪应力
8.  %                   4  --  第一主应力
9.  %                   5  --  第二主应力
10. %                   6  --  最大剪应力
11. %   返回值：
12. %       无
13.     global gNode gElement gNodeStress
14.
15.     switch iStress
16.     case 1
17.       title = 'x方向正应力' ;
18.     case 2
19.       title = 'y方向正应力' ;
20.     case 3
21.       title = '剪应力' ;
22.     case 4
23.       title = '最大主应力' ;
24.     case 5
25.       title = '最小主应力' ;
26.     case 6
27.       title = '最大剪应力' ;
28.     end
```

注释（第 2～12 行）

调整图形窗口标题（第 15～28 行）

```
29.
30.  figure;
31.  axis equal;
32.  axis off;
33.  set(gcf, '[illegible]', '[illegible]');
34.  set(gcf, '[illegible]', '[illegible]');
35.
36.  stressMin = [illegible](gNodeStress(:, [illegible]));
37.  stressMax = [illegible](gNodeStress(:, [illegible]));
38.  caxis([stre[illegible], stressMax]);
39.  colormap('[illegible]');
40.
41.  [el[illegible]_number, dummy] = size(gEl[illegible]);
42.  fo[illegible]1:[illegible]_number
43.    [illegible]
44.      [illegible]
45.      [illegible]
46.    [illegible]gNode[illegible]gElement(ie, 1), 2);
47.        gNode(gElement(ie, 2), 2);
48.        gNode(gElement(ie, 3), 2)];
49.    c = [gNodeStress(gElement(ie, 1), iStress);
50.        gNodeStress(gElement(ie, 2), iStress);
51.        gNodeStress(gElement(ie, 3), iStress)];
52.    set(patch(x, y, c), 'EdgeColor', 'interp');
53.  end
54.
55.  yTick = stressMin:(stressMax - stressMin)/10:stressMax;
56.  Label = cell(1, length(yTick));
57.  for i = 1:length(yTick)
58.      Label{i} = sprintf('%.2fMPa', yTick(i)/1e6);
59.  end
60.  set([illegible]olorbar('vert'), 'YTick', yTick, ...
61.      '[illegible]TickLabelMode', 'Manual', 'YTickLabel', Label);
62.  ret[illegible]
```

创建图形窗口，并[illegible]

[illegible]色映射表

绘制填充的三角形，显示应力云图

绘制颜色条，用来指示不同颜色所对应的应力值

图 4-22 和图 4-23 是用上述程序绘制的衬砌附近两个主应力的应力云图。如图 4-22 所示，在拱圈顶部附近和仰拱的拱底附近拉应力较大，边墙和中隔墙

基础底也出现较大拉应力。而最大的压应力出现在几个形状突变的区域，应是由于应力集中所致，另外左洞左拱脚亦有较大压应力出现，如图 **4-23** 所示。

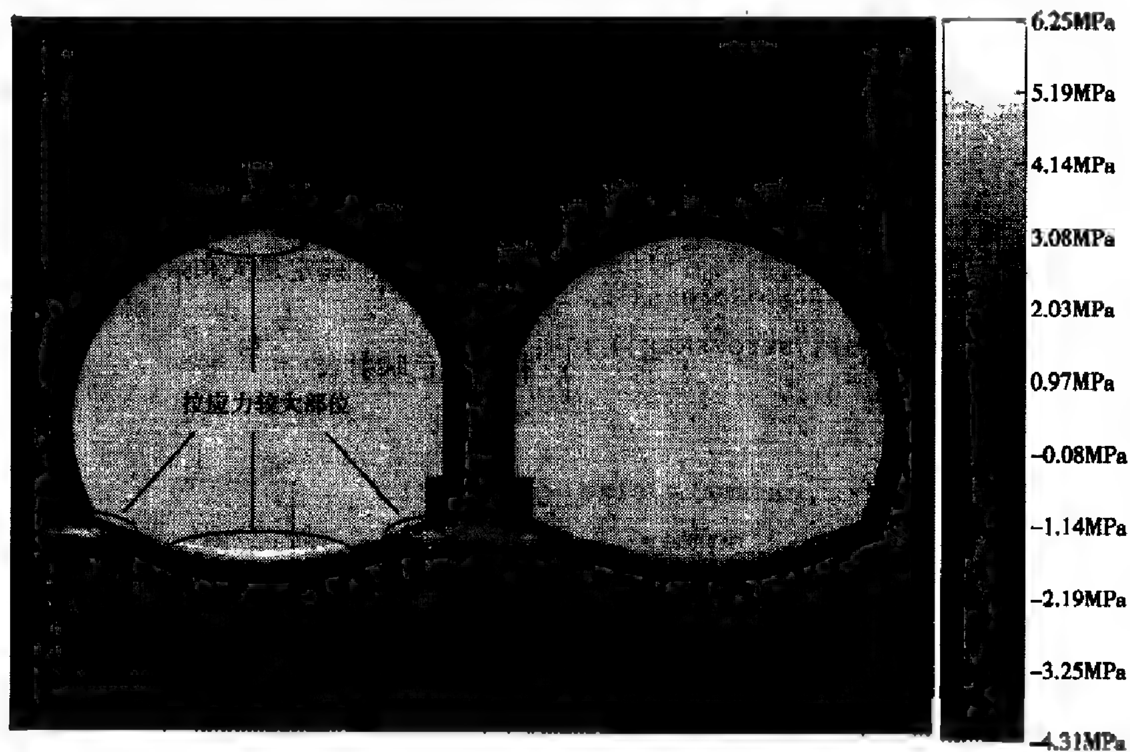


图 **4-22** 衬砌附近的第一主应力云图

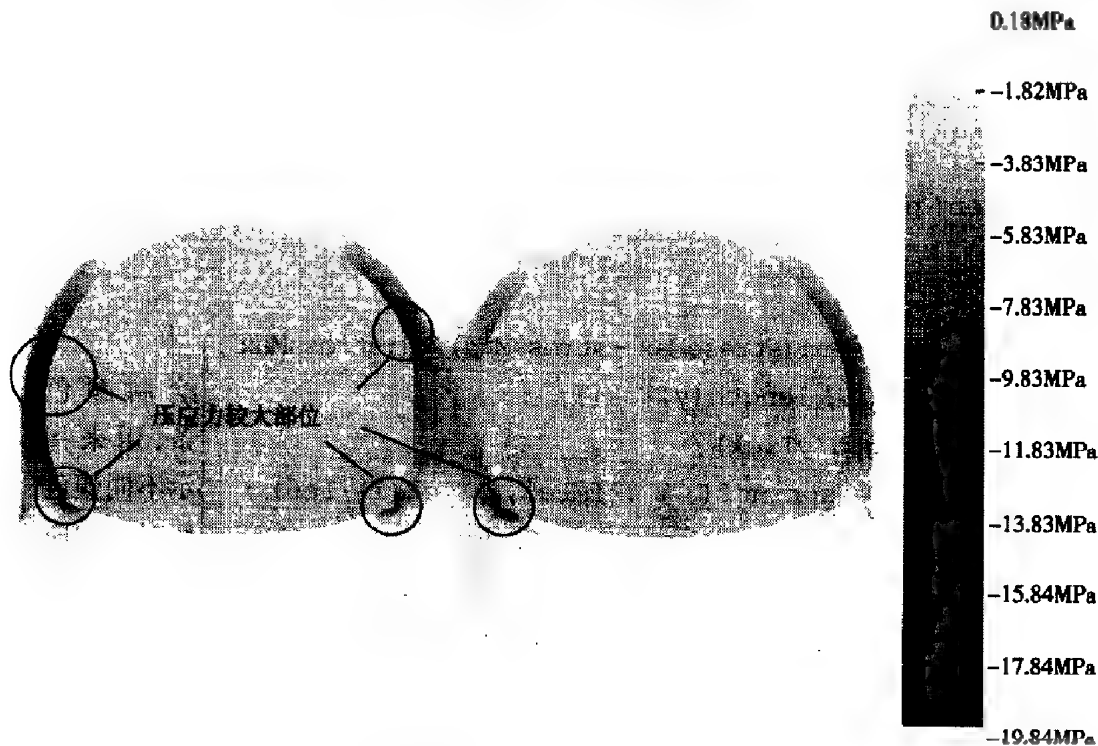


图 **4-23** 衬砌附近的第三主应力云图

参考文献

[1] Clough R. W. . The finite element in plane stress analysis. Proc. 2nd ASCE Conf. on Electronic Computation. Pittsburgh. Pa. . Sept,1960.

[2] Timoshenko S. P. . Goodier J. N. . Theory of elasticity. New York: McGraw-Hill,1970.

[3] 方德植. 解析几何. 北京:高等教育出版社,1986.

第5章 空间问题和空间轴称问题

空间三维实体单元可以认为是有限元中最一般的单元，因为所有的结构严格上来说都是三维的。尤其是在一些形状或受力十分复杂的区域，比如预应力混凝土结构的锚下区域，就既不能用应用平面假定，也无法把其近似为梁、板或壳结构。

本章介绍空间四面体单元及其线性位移模式。跟平面三角形单元相比，两者在概念及方法上都是相似的。另外，本章还介绍轴对称三角形单元，虽然轴对称问题在物理上是三维的，但在数学上是二维的，因此它与平面三角形单元有相似之处。所以这两种单元都可以看作是平面三角形单元的推广。

5.1 常应变四面体单元

本书第2章中已经给出了弹性力学空间问题的基本公式，有关它们的详细理论可以参考有关专著[1,2]。用有限元法计算弹性力学空间问题与平面问题相似，也是先划分单元。这里选择的单元类型是四面体单元，如图5-1所示，这种单元是由文献[3,4]提出的。设单元结点的编号为 i、j、m 和 p。

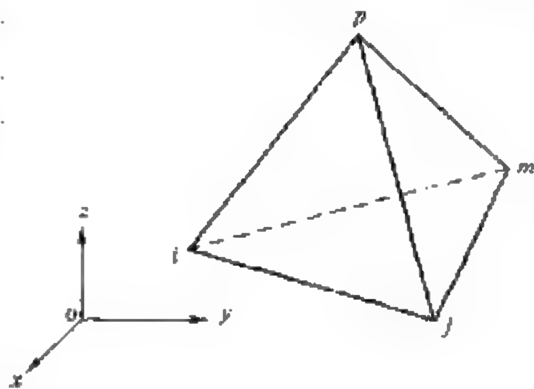


图5-1 四面体单元

在空间问题中，每个结点的位移有 3 个分量 u v 和 w，可以写成列阵形式

$$\boldsymbol{\delta}_i=\begin{Bmatrix}u_i\\v_i\\w_i\end{Bmatrix}\quad(i,j,m,p) \tag{5-1}$$

因此，整个单元共 12 个结点位移分量，可用列阵表示为

$$\boldsymbol{\delta}^e=[\boldsymbol{\delta}_i^T\quad\boldsymbol{\delta}_j^T\quad\boldsymbol{\delta}_m^T\quad\boldsymbol{\delta}_p^T]^T \tag{5-2}$$

与平面问题相似，假定单元内任一点的位移分量是坐标的线性函数

$$\begin{aligned}u&=\alpha_1+\alpha_2x+\alpha_3y+\alpha_4z\\v&=\alpha_5+\alpha_6x+\alpha_7y+\alpha_8z\\w&=\alpha_9+\alpha_{10}x+\alpha_{11}y+\alpha_{12}z\end{aligned} \tag{5-3}$$

设结点 i、j、m 和 p 的坐标分别为(x_i,y_i,z_i)、(x_j,y_j,z_j)、(x_m,y_m,z_m)和(x_p,y_p,z_p)，把它们代入式(5-3)的第一式，得出各结点在 x 方向的位移是

$$\begin{Bmatrix}u_i\\u_j\\u_m\\u_p\end{Bmatrix}=\begin{bmatrix}1&x_i&y_i&z_i\\1&x_j&y_j&z_j\\1&x_m&y_m&z_m\\1&x_p&y_p&z_p\end{bmatrix}\begin{Bmatrix}\alpha_1\\\alpha_2\\\alpha_3\\\alpha_4\end{Bmatrix} \tag{5-4}$$

解方程组(5-4)，求得 α_1、α_2、α_3 和 α_4 为

$$\alpha_1=\frac{1}{6V}\begin{vmatrix}u_i&x_i&y_i&z_i\\u_j&x_j&y_j&z_j\\u_m&x_m&y_m&z_m\\u_p&x_p&y_p&z_p\end{vmatrix}\qquad\alpha_2=\frac{1}{6V}\begin{vmatrix}1&u_i&y_i&z_i\\1&u_j&y_j&z_j\\1&u_m&y_m&z_m\\1&u_p&y_p&z_p\end{vmatrix}$$

$$\alpha_3=\frac{1}{6V}\begin{vmatrix}1&x_i&u_i&z_i\\1&x_j&u_j&z_j\\1&x_m&u_m&z_m\\1&x_p&u_p&z_p\end{vmatrix}\qquad\alpha_4=\frac{1}{6V}\begin{vmatrix}1&x_i&y_i&u_i\\1&x_j&y_j&u_j\\1&x_m&y_m&u_m\\1&x_p&y_p&u_p\end{vmatrix} \tag{5-5}$$

式中 V 是四面体单元 $ijmp$ 的体积[5,3]

$$6V=\begin{vmatrix}1 & x_i & y_i & z_i\\1 & x_j & y_j & z_j\\1 & x_m & y_m & z_m\\1 & x_p & y_p & z_p\end{vmatrix}\tag{5-6}$$

与平面三角形单元的面积类似，上面行列式值的符号依赖于四面体 4 个结点的排列顺序。当它们如图 5-1 所示的顺序排列时，上式的值是正的，否则为负。把式(5-5)回代到式(5-3)，得到用结点位移表示的单元内位移为

$$u=N_iu_i+N_ju_j+N_mu_m+N_pu_p\tag{5-7}$$

式中

$$\begin{aligned}N_i&=(a_i+b_ix+c_iy+d_iz)/6V\\N_j&=-(a_j+b_jx+c_jy+d_jz)/6V\\N_m&=(a_m+b_mx+c_my+d_mz)/6V\\N_p&=-(a_p+b_px+c_py+d_pz)/6V\end{aligned}\tag{5-8}$$

式中 a_i 和 $c_i(i,j,m,p)$ 分别是行列式(5-6)的第 1 和第 3 列各元素的余子式，而 b_i 和 $d_i(i,j,m,p)$ 则是第 2 和第 4 列各元素的余子式的负数，即

$$a_i=\begin{vmatrix}x_j & y_j & z_j\\x_m & y_m & z_m\\x_p & y_p & z_p\end{vmatrix}\quad b_i=-\begin{vmatrix}1 & y_j & z_j\\1 & y_m & z_m\\1 & y_p & z_p\end{vmatrix}$$

$$c_i=\begin{vmatrix}1 & x_j & z_j\\1 & x_m & z_m\\1 & x_p & z_p\end{vmatrix}\quad d_i=-\begin{vmatrix}1 & x_j & y_j\\1 & x_m & y_m\\1 & x_p & y_p\end{vmatrix}\qquad(i,j,m,p)\tag{5-9}$$

用同样的方法，对于另外两个位移分量也可以写成结点位移的形式

$$v=N_iv_i+N_jv_j+N_mv_m+N_pv_p\quad w=N_iw_i+N_jw_j+N_mw_m+N_pw_p\tag{5-10}$$

合并 3 个位移的表达式，可以写出单元内任意一点的位移列阵为

$$\boldsymbol{u}=\begin{Bmatrix}u\\v\\w\end{Bmatrix}=\begin{bmatrix}\boldsymbol{N}_i & \boldsymbol{N}_j & \boldsymbol{N}_m & \boldsymbol{N}_p\end{bmatrix}\boldsymbol{\delta}^e=\boldsymbol{N}\boldsymbol{\delta}^e\tag{5-11}$$

式中，形函数 $\boldsymbol{N}$ 的子矩阵为

$$\boldsymbol{N}_i=\begin{bmatrix} N_i & 0 & 0 \\ 0 & N_i & 0 \\ 0 & 0 & N_i \end{bmatrix}=N_i\boldsymbol{I}_3 \quad (i,j,m,p) \tag{5-12}$$

因为单元的位移都是结点位移的线性函数，因此在两个单元的交界面上位移是连续的。

已知单元内任意一点的位移后，就可确定单元体内任意一点的应变，把式(5-11)代入弹性力学几何方程式(2-47)，得到

$$\boldsymbol{\varepsilon}=\boldsymbol{L}^{\mathrm{T}}\boldsymbol{u}=\boldsymbol{L}^{\mathrm{T}}\boldsymbol{N}\boldsymbol{\delta}^e=\boldsymbol{B}\boldsymbol{\delta}^e=\begin{bmatrix}\boldsymbol{B}_i & -\boldsymbol{B}_j & \boldsymbol{B}_m & -\boldsymbol{B}_p\end{bmatrix}\boldsymbol{\delta}^e \tag{5-13}$$

式中

$$\boldsymbol{B}_i=\boldsymbol{L}^{\mathrm{T}}\boldsymbol{N}_i=\frac{1}{6V}\begin{bmatrix} b_i & 0 & 0 \\ 0 & c_i & 0 \\ 0 & 0 & d_i \\ 0 & d_i & c_i \\ d_i & 0 & b_i \\ c_i & b_i & 0 \end{bmatrix} \quad (i,j,m,p) \tag{5-14}$$

上式表明几何矩阵 $\boldsymbol{B}$ 中的元素都是常量，因此单元中的应变也都是常量。故采用线性位移模式的四面体单元是常应变单元。

按照本构关系(2-48)，得到应力矩阵

$$\boldsymbol{\sigma}=\boldsymbol{D}\boldsymbol{\varepsilon}=\boldsymbol{D}\boldsymbol{B}\boldsymbol{\delta}^e=\boldsymbol{S}\boldsymbol{\delta}^e=\begin{bmatrix}\boldsymbol{S}_i & -\boldsymbol{S}_j & \boldsymbol{S}_m & -\boldsymbol{S}_p\end{bmatrix}\boldsymbol{\delta}^e \tag{5-15}$$

把式(2-57)中的各向同性材料弹性矩阵 $\boldsymbol{D}$ 代入上式，得到应力矩阵的子矩阵为

$$\boldsymbol{S}_i=\boldsymbol{D}\boldsymbol{B}_i=\frac{A_3}{6V}\begin{bmatrix} b_i & A_1c_i & A_1d_i \\ A_1b_i & c_i & A_1d_i \\ A_1b_i & A_1c_i & d_i \\ 0 & A_2d_i & A_2c_i \\ A_2d_i & 0 & A_2b_i \\ A_2c_i & A_2b_i & 0 \end{bmatrix} \quad (i,j,m,p) \tag{5-16}$$

式中 $$A_1=\frac{\mu}{1-\mu}\quad A_2=\frac{1-2\mu}{2(1-\mu)}\quad A_3=\frac{E(1-\mu)}{(1+\mu)(1-2\mu)}\tag{5-17}$$

显然,单元中的应力也是常量。因此,跟常应变三角形单元一样,应变和应力在单元间是不连续的。同样，可以采用绕结点平均法计算结点上的应力值。

5.2 四面体的体积坐标

与三角形的面积坐标类似，也可以在四面体单元上定义体积坐标。四面体单元内点 q 的体积坐标 S_i、S_j、S_m、S_p 的定义为

$$S_i=V_i/V\quad(i,j,m,p)\tag{5-18}$$

式中 V_i 是四面体 $qjmp$ 的体积，V_j 是四面体 $qimp$ 的体积,……如图 5-2 所示。

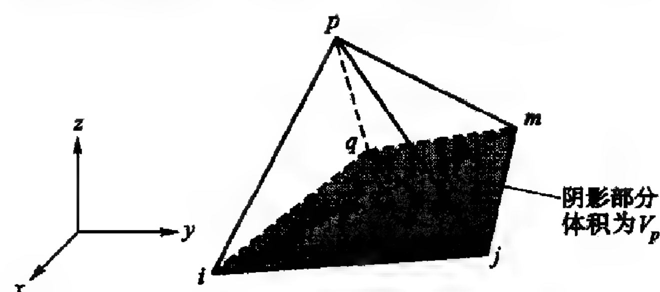


图 5-2 体积坐标

设点 q 在直角坐标系下的坐标为(x,y,z),则有

$$V_i=\frac{1}{6}\begin{vmatrix}1 & x & y & z\\ 1 & x_j & y_j & z_j\\ 1 & x_m & y_m & z_m\\ 1 & x_p & y_p & z_p\end{vmatrix}\tag{5-19}$$

将行列式(5-19)按第一行展开,并应用式(5-9),得到

$$V_i=\frac{1}{6}(a_i+b_ix+c_iy+d_iz)\quad(i,m)\tag{5-20}$$

类似地可以得到另外两个小四面体的体积为

$$V_j=-\frac{1}{6}(a_j+b_jx+c_jy+d_jz)\quad(j,p)\tag{5-21}$$

因此

$$S_i = \frac{1}{6V}(a_i + b_i x + c_i y + d_i z) \qquad (i,m) \tag{5-22}$$

$$S_j = -\frac{1}{6V}(a_j + b_j x + c_j y + d_j z) \qquad (j,p) \tag{5-23}$$

比较式(5-22)、式(5-23)和式(5-8)很容易发现四面体的体积坐标就是四面体单元的形函数。

因为 $V = V_i + V_j + V_m + V_p$,所以

$$S_i + S_j + S_m + S_p = 1 \tag{5-24}$$

因此在4个体积坐标中,只有3个是独立的。公式(5-22)和式(5-23)是从直角坐标到体积坐标的变换关系。通过求逆,可以得到相反的变换式,为此,把式(5-22)和式(5-23)写成矩阵形式

$$\begin{Bmatrix} S_i \\ S_j \\ S_m \\ S_p \end{Bmatrix} = \frac{1}{6V}\begin{bmatrix} a_i & b_i & c_i & d_i \\ -a_j & -b_j & -c_j & -d_j \\ a_m & b_m & c_m & d_m \\ -a_p & -b_p & -c_p & -d_p \end{bmatrix}\begin{Bmatrix} 1 \\ x \\ y \\ z \end{Bmatrix} = \frac{1}{6V}\boldsymbol{A}\begin{Bmatrix} 1 \\ x \\ y \\ z \end{Bmatrix} \tag{5-25}$$

根据式(5-9)的 a_i、b_i、c_i 和 $d_i(i,j,m,p)$和线性代数中伴随矩阵的定义,矩阵 $\boldsymbol{A}$ 的逆矩阵就是

$$\boldsymbol{A}^{-1} = 6V\begin{bmatrix} 1 & 1 & 1 & 1 \\ x_i & x_j & x_m & x_p \\ y_i & y_j & y_m & y_p \\ z_i & z_j & z_m & z_p \end{bmatrix} \tag{5-26}$$

所以从体积坐标到直角坐标的变换关系为

$$\begin{Bmatrix} 1 \\ x \\ y \\ z \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_i & x_j & x_m & x_p \\ y_i & y_j & y_m & y_p \\ z_i & z_j & z_m & z_p \end{bmatrix}\begin{Bmatrix} S_i \\ S_j \\ S_m \\ S_p \end{Bmatrix} \tag{5-27}$$

利用体积坐标中进行单元分析时,要用到关于微分和积分的坐标变换。对直角坐标的导数,不难由复合函数求导数公式得到,例如

$$\frac{\partial}{\partial x} = \sum_{k=i,j,m,p}\frac{\partial}{\partial S_k}\frac{\partial S_k}{\partial x} = \frac{1}{6V}\sum_{k=i,j,m,p} b_k \frac{\partial}{\partial S_k} \tag{5-28}$$

关于以体积坐标表达的函数在四面体单元上的积分，可以有如下形式的积分公式利用

$$\iiint_V S_i^{m_1} S_j^{m_2} S_m^{m_3} S_p^{m_4} \mathrm{d}V = \frac{m_1!\,m_2!\,m_3!\,m_4!}{(m_1 + m_2 + m_3 + m_4 + 3)!} 6V \tag{5-29}$$

当计算表面力的等效结点力时，常要计算在四面体的某个面上对体积坐标进行积分，比如在(i,j,m)3个结点组成的三角形面上，可以应用下面的积分公式

$$\iint_{\Delta_{ijm}} S_i^{\alpha} S_j^{\beta} S_m^{\gamma} \mathrm{d}A = \frac{\alpha!\,\beta!\,\gamma!}{(\alpha + \beta + \gamma + 2)!} 2\Delta_{ijm} \tag{5-30}$$

式中，Δ_{ijm}是(i,j,m)3个结点组成的三角形的面积。这里我们发现，这个公式跟三角形的面积积分公式完全一致。实际上，四面体的体积坐标在四面体的某一个面上就退化为该面上的面积坐标。

5.3 四面体单元的刚度矩阵和等效结点力

利用刚度矩阵的普遍公式(2-70)，并注意到应变矩阵在单元内为常数矩阵的事实，得到空间问题四面体单元的刚度矩阵公式

$$\boldsymbol{K}^e = \iiint_V \boldsymbol{B}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{B} \mathrm{d}x\mathrm{d}y\mathrm{d}z = \boldsymbol{B}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{B} V \tag{5-31}$$

或写成分块形式

$$\boldsymbol{K}^e = \begin{bmatrix} \boldsymbol{K}_{ii} & -\boldsymbol{K}_{ij} & \boldsymbol{K}_{im} & \boldsymbol{K}_{ip} \\ -\boldsymbol{K}_{ji} & \boldsymbol{K}_{jj} & -\boldsymbol{K}_{jm} & \boldsymbol{K}_{jp} \\ \boldsymbol{K}_{mi} & -\boldsymbol{K}_{mj} & \boldsymbol{K}_{mm} & -\boldsymbol{K}_{mp} \\ -\boldsymbol{K}_{pi} & \boldsymbol{K}_{pj} & -\boldsymbol{K}_{pm} & \boldsymbol{K}_{pp} \end{bmatrix} \tag{5-32}$$

式中子矩阵

$$\boldsymbol{K}_{rs} = \boldsymbol{B}_r^{\mathrm{T}} \boldsymbol{D} \boldsymbol{B}_s V \qquad (r,s = i,j,m,p) \tag{5-33}$$

把$\boldsymbol{B}$的表达式(5-14)和各向同性材料的弹性矩阵$\boldsymbol{D}$的表达式(2-57)代入，得到$\boldsymbol{K}_{rs}$的表达式如下

$$\boldsymbol{K}_{rs} = \frac{A_3}{36V} \begin{bmatrix} b_r b_s + A_2(c_r c_s + d_r d_s) & A_1 b_r c_s + A_2 c_r b_s & A_1 b_r d_s + A_2 d_r b_s \\ A_1 c_r b_s + A_2 b_r c_s & c_r c_s + A_2(d_r d_s + b_r b_s) & A_1 c_r d_s + A_2 d_r c_s \\ A_1 d_r b_s + A_2 b_r d_s & A_1 d_r c_s + A_2 c_r d_s & d_r d_s + A_2(b_r b_s + c_r c_s) \end{bmatrix}$$

$$(r,s = i,j,m,p) \tag{5-34}$$

上式的推导过程就是 3 个矩阵相乘，利用 MATLAB 的符号运算功能非常容易求得。

四结点四面体单元的单元刚度矩阵 K^e 是 12 阶的方阵，每一个元素是由单元结点的坐标和单元材料的弹性常数所决定，所以是一个常量矩阵。

下面是计算四面体单元刚度矩阵的程序

```
1.  function k = StiffnessMatrix( ie )
2.  %  计算单元刚度矩阵
3.  %  输入参数:
4.  %      ie ---- 单元号
5.  %  返回值:
6.  %      k ---- 单元刚度矩阵
7.
8.  global gNode gElement gMaterial
9.
10.     % 获取结点坐标
11.     x = gNode( gElement( ie, : ), 1 );
12.     y = gNode( gElement( ie, : ), 2 );
13.     z = gNode( gElement( ie, : ), 3 );
14.
15.     % 计算 6V
16.     V6 = det( [1 x(1) y(1) z(1)
17.                1 x(2) y(2) z(2)
18.                1 x(3) y(3) z(3)
19.                1 x(4) y(4) z(4) ] );
20.
21.     % 计算应变矩阵
22.     B = zeros( 6, 12 );
23.     for i=1:4
24.         j = mod(i, 4) + 1;
25.         m = mod(i+1,4) + 1;
26.         p = mod(i+2,4) + 1;
27.         bi = - det([1 y(j) z(j)
28.                     1 y(m) z(m)
29.                     1 y(p) z(p) ]);
```

```
30.        ci = det([ 1 x(j) z(j)
31.                   1 x(m) z(m)
32.                   1 x(p) z(p) ] );
33.        di = - det([ 1 x(j) y(j)
34.                     1 x(m) y(m)
35.                     1 x(p) y(p) ] );
36.        B(:,(i-1)*3+1:(i-1)*3+3) = (-1)^(i+1)*[bi  0  0
37.                                                0  ci  0
38.                                                0  0  di
39.                                                0  di  ci
40.                                                di  0  bi
41.                                                ci  bi  0];
42.    end
43.    B = B / V6 ;
44.
45.    % 计算弹性矩阵
46.    E = gMaterial( gElement( ie, 5 ), 1 ) ;
47.    mu = gMaterial( gElement( ie, 5 ), 2 ) ;
48.    A1 = mu/(1-mu) ;
49.    A2 = (1-2*mu)/2/(1-mu) ;
50.    A3 = E*(1-mu)/(1+mu)/(1-2*mu) ;
51.    D = [  1  A1  A1  0  0  0
52.          A1   1  A1  0  0  0
53.          A1  A1   1  0  0  0
54.           0   0   0 A2  0  0
55.           0   0   0  0 A2  0
56.           0   0   0  0  0 A2];
57.    D = D * A3 ;
58.
59.    % 计算单元刚度矩阵
60.    V = abs(V6)/6 ;
61.    k = transpose( B ) * D * B * V ;
62. return
```

利用单元等效结点力的普遍公式(2-72),可计算作用于四面体单元上的各种外荷载分配到结点上的等效结点力。将单元刚度矩阵和等效结点力之

后，就可按照第二章的思路和方法，对单元刚度矩阵和等效结点力进行扩大叠加，得到整体刚度矩阵和结点力向量。

下面先来分析均质单元自重的等效结点力。设重力沿 z 轴的负方向，则根据式(2-72)，并利用积分公式(5-29)得到每一个结点 z 方向的等效结点力为

$$f_{zi} = -\rho g \iiint_V N_i \mathrm{d}V = -\frac{1}{4}\rho g V = -\frac{1}{4} W \quad (i,j,m,p) \tag{5-35}$$

上式表明，将均质单元的自重 W 平均分配到 4 个结点，即为等效结点力。

分布面力在工程中也很常见。设单元的某一表面 ijm，承受线性分布荷载，它在 i、j 和 m 三个结点处的强度分别是 p_{si}、p_{sj} 和 p_{sm}，则根据式(2-72)，分配到结点 i、j 和 m 上的等效结点力的数值分别为

$$f_{si} = \iint_{\Delta_{ijm}} N_i (N_i p_{si} + N_j p_{sj} + N_m p_{sm}) \mathrm{d}A \quad (i,j,m) \tag{5-36}$$

利用积分公式(5-30)，得到

$$f_{si} = \frac{1}{6}\left(p_{si} + \frac{1}{2} p_{sj} + \frac{1}{2} p_{sm}\right)\Delta_{ijm} \quad (i,j,m) \tag{5-37}$$

式中结点力的方向与原分布荷载的方向平行。这里要用到三维空间中三角形的面积，根据解析几何[5]，有

$$\Delta_{ijm} = \frac{1}{2} \left| \vec{ij} \times \vec{im} \right| \quad (i,j,m,p) \tag{5-38}$$

式中 $\vec{ij}$ 和 $\vec{im}$ 分别表示从结点 i 到 j 和 m 的两个矢量。展开即为

$$\Delta_{ijm} = \frac{1}{2}\sqrt{b_p^2 + c_p^2 + d_p^2} \quad (i,j,m,p) \tag{5-39}$$

式中 b_p、c_p 和 d_p 的定义见式(5-9)。当分布面力是垂直于表面上的压力时，应用式(5-37)后，还必须把它们转换到沿坐标轴方向，才能叠加到整体结点力向量中。如果定义从三角形 ijm 指向四面体另一个顶点 p 的方向为该三角形的法线方向，则由解析几何知，法线 n 的 3 个方向余弦为

$$n_x = -\frac{1}{2\Delta_{ijm}} b_p, \quad n_y = -\frac{1}{2\Delta_{ijm}} c_p, \quad n_z = -\frac{1}{2\Delta_{ijm}} d_p \tag{5-40}$$

同样式中 b_p、c_p 和 d_p 由式(5-9)定义。上式也是作用在三角形 ijm 上压力 p 的

方向,因此沿坐标轴的 3 个分量是

$$f_{sxi} = n_x f_{si}, \quad f_{syi} = n_y f_{si} \qquad f_{szi} = n_z f_{si} \tag{5-41}$$

计算线性分布压力的等效结点力的 MATLAB 程序如下:

```
1.  function enf = EquivalentDistPressure( node, pressure )
2.  %    计算线性分布压力的等效结点力
3.  %    输入参数:
4.  %   node -----------  结点号
5.  %   pressure -------  与结点号对应的压力值
6.  %    返回值:
7.  %   enf -----------  等效结点力向量
8.  global gNode
9.
10.  enf = zeros( 9, 1 ) ;
11.  % 计算作用压力的三角形的面积
12.  x = gNode( node, 1 ) ;
13.  y = gNode( node, 2 ) ;
14.  z = gNode( node, 3 ) ;
15.  b = - det( [1  y(1)  z(1)
16.              1  y(2)  z(2)
17.              1  y(3)  z(3)] ) ;
18.  c =   det( [1  x(1)  z(1)
19.              1  x(2)  z(2)
20.              1  x(3)  z(3)] ) ;
21.  d = - det( [1  x(1)  y(1)
22.              1  x(2)  y(2)
23.              1  x(3)  y(3)] ) ;
24.  A2 = sqrt( b^2 + c^2 + d^2 ) ;
25.  A = A2 / 2 ;
26.
27.  % 计算 3 个结点上的等效结点力
28.  f1 = ( pressure(1) + pressure(2)/2 + pressure(3)/2 ) / 6 * A ;
29.  f2 = ( pressure(1)/2 + pressure(2) + pressure(3)/2 ) / 6 * A ;
30.  f3 = ( pressure(1)/2 + pressure(2)/2 + pressure(3)) / 6 * A ;
31.
```

```
32.  % 计算3个方向上的分量
33.  nx = b/A2;
34.  ny = c/A2;
35.  nz = -d/A2;
36.  enf(1:3) = f1*[nx;ny;nz];
37.  enf(4:6) = f2*[nx;ny;nz];
38.  enf(7:9) = f3*[nx;ny;nz];
39.  return
```

最后来考察由于温度改变在单元上引起的等效结点力

$$\boldsymbol{f}_T = \iiint_V \boldsymbol{B}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{\varepsilon}_0 \mathrm{d}x\mathrm{d}y\mathrm{d}z \tag{5-42}$$

对于各向同性材料，有

$$\boldsymbol{\varepsilon}_0 = \alpha T \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}} \tag{5-43}$$

对于四面体单元，由于矩阵 $\boldsymbol{B}$ 的元素都是常量，上式可以写成

$$\boldsymbol{f}_T = \alpha \boldsymbol{B}^{\mathrm{T}} \boldsymbol{D} \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}} \iiint_V T \mathrm{d}x\mathrm{d}y\mathrm{d}z \tag{5-44}$$

若温度分布采用线性模式，则上式的积分

$$\iiint_V T \mathrm{d}x\mathrm{d}y\mathrm{d}z = \frac{1}{4}(T_i + T_j + T_m + T_p)V \tag{5-45}$$

式中，T_i、T_j、T_m 和 T_p 为结点 i、j、m 和 p 处的温度改变量。将 $\boldsymbol{B}$、$\boldsymbol{D}$ 的表达式及式（5-45）代入式（5-44），得到四面体单元在温度改变中引起的等效结点力为

$$\boldsymbol{f}_T = \frac{E\alpha(T_i + T_j + T_m + T_p)}{24(1-2\mu)} \times \begin{bmatrix} b_i & c_i & d_i & -b_j & -c_j & -d_j & b_m & c_m & d_m & -b_p & -c_p & -d_p \end{bmatrix}^{\mathrm{T}} \tag{5-46}$$

5.4 空间轴对称问题与三角形截面环单元

如果空间结构的几何形状、约束条件及荷载分布都对称于某个轴，则其位移、应变和应力等也对称于该轴而与周向坐标无关，这种情形称为空间轴对称

问题。如在圆形分布荷载作用下的半无限地基是典型的轴对称问题[6,7]。一般习惯采用圆柱坐标系(r,θ,z)描述空间轴对称问题，其中 z 轴为对称轴。这样结构内的位移、应变和应力只是 r 和 z 的函数，而与 θ 无关，因此数学上是二维问题。最简单的轴对称单元是三角形截面的环单元[8,9]，它是由在 rz 平面上的三角形 ijm 绕对称轴 z 回转一周得到的。单元的棱边都是圆，称为结圆，每个结圆与 rz 平面的交点就是结点，如图 5-3 所示。这样，轴对称弹性体在 rz 平面上的截面将被各单元划分成三角形网格，就像平面问题中各三角形单元在 xy 平面上形成的三角形网格一样。又因为在轴对称问题中，物体内任意一点只有径向位移 u 和轴向位移 w，并且它们仅与坐标 r 和 z 有关，而与 θ 无关。因此，像平面问题一样，只需考虑坐标平面 rz 上的网格，取结点位移为基本未知量，单元结点位移可用列阵表示为

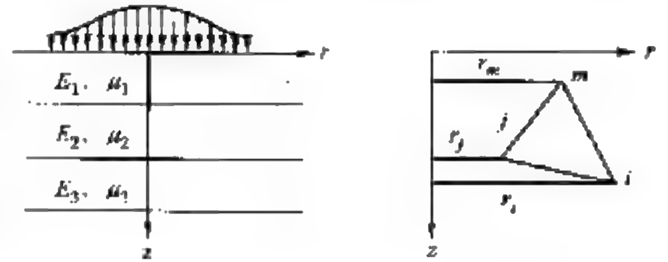


图 5-3　层状地基的轴对称假定和三角形环单元

$$\boldsymbol{\delta}^e = [\boldsymbol{\delta}_i^{\mathrm{T}} \quad \boldsymbol{\delta}_j^{\mathrm{T}} \quad \boldsymbol{\delta}_m^{\mathrm{T}}]^{\mathrm{T}} = [u_i \quad w_i \quad u_j \quad w_j \quad u_m \quad w_m]^{\mathrm{T}} \tag{5-47}$$

仿照平面问题，取线性位移模式

$$\begin{aligned} u &= \alpha_1 + \alpha_2 r + \alpha_3 z \\ w &= \alpha_4 + \alpha_5 r + \alpha_6 z \end{aligned} \tag{5-48}$$

必然得到与平面问题中相似的结果，单元内任意一点的位移为

$$\begin{aligned} u &= N_i u_i + N_j u_j + N_m u_m \\ w &= N_i w_i + N_j w_j + N_m w_m \end{aligned} \tag{5-49}$$

其中形函数

$$N_i = (a_i + b_i r + c_i z)/2\Delta \qquad (i,j,m) \tag{5-50}$$

而

$$\Delta = \frac{1}{2}\begin{vmatrix} 1 & r_i & z_i \\ 1 & r_j & z_j \\ 1 & r_m & z_m \end{vmatrix} \tag{5-51}$$

$$a_i = \begin{vmatrix} r_j & z_j \\ r_m & z_m \end{vmatrix} \quad b_i = -\begin{vmatrix} 1 & z_j \\ 1 & z_m \end{vmatrix} \quad c_i = \begin{vmatrix} 1 & r_j \\ 1 & r_m \end{vmatrix} \quad (i,j,m) \tag{5-52}$$

式(5-49)也可写成矩阵形式

$$\boldsymbol{u} = [u \quad w]^{\mathrm{T}} = \boldsymbol{N}\boldsymbol{\delta}^e = [\boldsymbol{N}_i \quad \boldsymbol{N}_j \quad \boldsymbol{N}_m]\boldsymbol{\delta}^e \tag{5-53}$$

式中形函数的子矩阵为

$$\boldsymbol{N}_i = \begin{bmatrix} N_i & 0 \\ 0 & N_i \end{bmatrix} = N_i \boldsymbol{I}_2 \tag{5-54}$$

将式(5-49)代入轴对称问题的几何方程得到单元内任意一点的应变为

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_r \\ \varepsilon_\theta \\ \varepsilon_z \\ \gamma_{rz} \end{Bmatrix} = \begin{Bmatrix} \partial u/\partial r \\ u/r \\ \partial w/\partial z \\ \partial w/\partial r + \partial u/\partial z \end{Bmatrix} = \frac{1}{2\Delta}\begin{bmatrix} b_i & 0 & b_j & 0 & b_m & 0 \\ g_i & 0 & g_j & 0 & g_m & 0 \\ 0 & c_i & 0 & c_j & 0 & c_m \\ c_i & b_i & c_j & b_j & c_m & b_m \end{bmatrix}\begin{Bmatrix} u_i \\ w_i \\ u_j \\ w_j \\ u_m \\ w_m \end{Bmatrix} \tag{5-55}$$

式中

$$g_i = a_i/r + b_i + c_i z/r \tag{5-56}$$

式(5-55)可以简写为

$$\boldsymbol{\varepsilon} = \boldsymbol{B}\boldsymbol{\delta}^e \tag{5-57}$$

式中

$$\boldsymbol{B} = [\boldsymbol{B}_i \quad \boldsymbol{B}_j \quad \boldsymbol{B}_m], \quad \boldsymbol{B}_i = \frac{1}{2\Delta}\begin{bmatrix} b_i & 0 \\ g_i & 0 \\ 0 & c_i \\ c_i & b_i \end{bmatrix} \quad (i,j,m) \tag{5-58}$$

由此可见，单元中的应变分量 ε_r、ε_z 和 γ_{rz} 都是常量，但是环向正应变 ε_θ 中包含有 r 和 z 不是常量。单元内任意一点的应力分量可表示为

$$\boldsymbol{\sigma}=\begin{Bmatrix}\sigma_r\\ \sigma_\theta\\ \sigma_z\\ \tau_{rz}\end{Bmatrix}=\boldsymbol{D\varepsilon}=\boldsymbol{DB\delta}^e=\boldsymbol{S\delta}^e=\begin{bmatrix}\boldsymbol{S}_i & \boldsymbol{S}_j & \boldsymbol{S}_m\end{bmatrix}\boldsymbol{\delta}^e \tag{5-59}$$

对于各向同性材料，弹性矩阵为

$$\boldsymbol{D}=A_3\begin{bmatrix}1 & A_1 & A_1 & 0\\ A_1 & 1 & A_1 & 0\\ A_1 & A_1 & 1 & 0\\ 0 & 0 & 0 & A_2\end{bmatrix} \tag{5-60}$$

式中的 A_1、A_2 和 A_3 由式(5-17)定义，则应力矩阵是

$$\boldsymbol{S}_i=\frac{A_3}{2\Delta}\begin{bmatrix}b_i+A_1g_i & A_1c_i\\ A_1b_i+g_i & A_1c_i\\ A_1(b_i+g_i) & c_i\\ A_2c_i & A_2b_i\end{bmatrix}\quad(i,j,m) \tag{5-61}$$

显然，只有应力分量 τ_{rz} 在单元中是常量外，其余3个正应力在单元中都不是常量。在使用上，为了简化计算和消除对称轴上由于 $r=0$ 所引起的麻烦，常把各个单元中的 r 及 z 用单元形心的坐标近似，即

$$r\approx\bar{r}=\frac{1}{3}(r_i+r_j+r_m)\quad z\approx\bar{z}=\frac{1}{3}(z_i+z_j+z_m) \tag{5-62}$$

于是式(5-56)成为

$$g_i\approx\bar{g}_i=\frac{a_i}{\bar{r}}+b_i+\frac{c_i\bar{z}}{\bar{r}}\quad(i,j,m) \tag{5-63}$$

这样，就可把各个单元近似地当作常应变单元。有意思的是，经过这样的近似处理，可以验证 $\bar{g}_i=\bar{g}_j=\bar{g}_m$。

5.5 轴对称单元的刚度矩阵和等效结点力

轴对称问题的单元刚度矩阵可以利用平面问题的刚度矩阵公式列出。但

是要注意它与平面问题三角形单元的情况稍有不同。因为轴对称问题中采用的是三角形截面的环单元，积分应该在整个环内进行，故有

$$\boldsymbol{K}^e = \iiint_V \boldsymbol{B}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{B} \mathrm{d}V = \iiint_V \boldsymbol{B}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{B} r \mathrm{d}r \mathrm{d}z \mathrm{d}\theta \tag{5-64}$$

又由于被积函数与 θ 无关，故积分仍可视为在 rz 平面上的三角形环单元的截面内进行，而在积分号外乘上 2π，于是

$$\boldsymbol{K}^e = 2\pi \iint_\Delta \boldsymbol{B}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{B} r \mathrm{d}r \mathrm{d}z \tag{5-65}$$

把式(5-58)代入上式，并写成分块形式

$$\boldsymbol{K}^e = \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ij} & \boldsymbol{K}_{im} \\ \boldsymbol{K}_{ji} & \boldsymbol{K}_{jj} & \boldsymbol{K}_{jm} \\ \boldsymbol{K}_{mi} & \boldsymbol{K}_{mj} & \boldsymbol{K}_{mm} \end{bmatrix} \tag{5-66}$$

其中子矩阵为

$$\boldsymbol{K}_{st} = 2\pi \iint_\Delta \boldsymbol{B}_s^{\mathrm{T}} \boldsymbol{D} \boldsymbol{B}_t r \mathrm{d}r \mathrm{d}z \quad (s,t = i,j,m) \tag{5-67}$$

由于在轴对称问题的矩阵 $\boldsymbol{B}$ 中出现坐标 r 和 z，所以上式的积分运算比平面问题要复杂。如果取单元形心的坐标 $\bar{r}$ 和 $\bar{z}$ 替代矩阵 $\boldsymbol{B}$ 中的坐标 r 和 z 作为一次近似，得到一个近似的单元刚度矩阵。这样，式5-67 可简化为

$$\boldsymbol{K}_{st} = 2\pi \boldsymbol{B}_s^{\mathrm{T}} \boldsymbol{D} \boldsymbol{B}_t \bar{r} \Delta \tag{5-68}$$

写成显式为

$$\boldsymbol{K}_{st} = \frac{\pi \bar{r} A_3}{2\Delta} \begin{bmatrix} b_s(b_t + A_1 \bar{g}_t) + \bar{g}_s(\bar{g}_t + A_1 b_t) + A_2 c_s c_t & A_1 c_t(b_s + \bar{g}_s) + A_2 c_s b_t \\ A_1 c_s(b_t + \bar{g}_t) + A_2 b_s c_t & c_s c_t + A_2 b_s b_t \end{bmatrix}$$

$$(s,t = i,j,m) \tag{5-69}$$

参考文献[10,11]给出了精确的单元刚度矩阵，并进行具体算例的比较，认为在单元不大的情况下，近似刚度矩阵导致的误差在5%以内。

这里给出了计算近似单元刚度矩阵的子程序。

```
1. function k = StiffnessMatrix( ie )
2. %    计算单元刚度矩阵
3. %    输入参数：
4. %        ie ----  单元号
```

```
5.  %  返回值:
6.  %      k  ---- 单元刚度矩阵
7.
8.      global gNode gElement gMaterial
9.      E  = gMaterial( gElement(ie, 4), 1 ) ;
10.     mu = gMaterial( gElement(ie, 4), 2 ) ;
11.     ri = gNode( gElement( ie, 1 ), 1 ) ;
12.     zi = gNode( gElement( ie, 1 ), 2 ) ;
13.     rj = gNode( gElement( ie, 2 ), 1 ) ;
14.     zj = gNode( gElement( ie, 2 ), 2 ) ;
15.     rm = gNode( gElement( ie, 3 ), 1 ) ;
16.     zm = gNode( gElement( ie, 3 ), 2 ) ;
17.     ai = rj* zm - rm* zj ;
18.     aj = rm* zi - ri* zm ;
19.     am = ri* zj - rj* zi ;
20.     bi = zj - zm ;
21.     bj = zm - zi ;
22.     bm = zi - zj ;
23.     ci = - (rj - rm) ;
24.     cj = - (rm - ri) ;
25.     cm = - (ri - rj) ;
26.     area = (ai + aj + am) /2 ;
27.     r = (ri + rj + rm) /3;
28.     z = (zi + zj + zm) /3 ;
29.     gi = ai/r + bi + ci* z/r ;
30.     gj = aj/r + bj + cj* z/r ;
31.     gm = am/r + bm + cm* z/r ;
32.     B = [bi  0 bj  0 bm  0
33.          gi  0 gj  0 gm  0
34.          0 ci  0 cj  0 cm
35.          ci bi cj bj cm bm] ;
36.   B = B/2/area ;
37.   A1 = mu/(1 -mu) ;
38.   A2 = (1 - 2* mu)/2/(1 -mu) ;
39.   A3 = E* (1 -mu)/(1 +mu)/(1 -2* mu) ;
40.   D = A3* [ 1   A1   A1         0
```

```
41.  A1  1  A1     0
42.  A1  A1  1     0
43.  0   0   0   A2];
44.  k = 2* pi* r* abs(area)* transpose(B)* D* B;
45.  return
```

轴对称问题的等效结点力可利用普遍公式(2-72)写出。对作用于三角形环单元上的体积力、表面力也应该都是轴对称的,因此它们的等效结点力可写成为

$$f_v^e = 2\pi\iint_\Delta N^T p_v r \mathrm{d}r\mathrm{d}z \tag{5-70}$$

$$f_s^e = 2\pi\int_s N^T p_s r \mathrm{d}s \tag{5-71}$$

把上面两式与平面问题三角形单元的等效结点力相比，可以发现在轴对称情况中被积函数比平面问题的多一个变量 r,所以虽然也是采用线性位移模式，但是不能利用平面问题的公式来求等效结点力。当然我们仍然可以应用平面问题三角形单元的面积坐标积分公式(4-45)和式(4-46)。如果体积力或表面力可表示为坐标 r 和 z 的多项式，则不难利用面积坐标的积分公式得到显式的等效结点力。

体积力中最常见的是单元的自重，此时 $p_{vr}=0, p_{vz}=-\rho g$,其中 ρ 为密度,g 是重力加速度。于是单元的自重分配到结点上的等效结点力为

$$f_{vi}^e = \begin{Bmatrix} f_{vri} \\ f_{vzi} \end{Bmatrix} = 2\pi\iint_\Delta N_i \begin{Bmatrix} 0 \\ -\rho g \end{Bmatrix} r\mathrm{d}r\mathrm{d}z \quad (i,j,m) \tag{5-72}$$

为了利用面积坐标的积分公式，先把径向坐标 r 写成

$$r = r_i L_i + r_j L_j + r_m L_m \tag{5-73}$$

然后代入式(5-72),并完成积分,就得到

$$f_{vi}^e = \begin{Bmatrix} f_{vri} \\ f_{vzi} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -\dfrac{\pi\rho\Delta}{6}(3\bar{r}+r_i) \end{Bmatrix} \quad (i,j,m) \tag{5-74}$$

对于轴对称的机械结构，离心力也是很常见的，它也是一种体积力。此时,$p_{vr}=\rho\omega^2 r, p_{vz}=0$,其中 ω 是角速度。于是单元的离心力分配到结点上的等效结点力为

$$f_{vi}^e = \begin{Bmatrix} f_{vri} \\ f_{vzi} \end{Bmatrix} = 2\pi\iint_\Delta N_i \begin{Bmatrix} \rho\omega^2 r \\ 0 \end{Bmatrix} r\mathrm{d}r\mathrm{d}z \quad (i,j,m) \tag{5-75}$$

同样利用式(5-73)和面积坐标的积分公式得到

$$f_{\omega i}^{e}=\begin{Bmatrix}f_{\omega ri}\\f_{\omega zi}\end{Bmatrix}=\begin{Bmatrix}\dfrac{\pi\rho\omega^{2}\Delta}{15}(9\bar{r}^{2}+2r_{i}^{2}-r_{j}r_{m})\\0\end{Bmatrix}\qquad(i,j,m)\tag{5-76}$$

线性分布的压力是最常见的表面力。设在 rz 平面上单元 ijm 的 ij 边上受到线性分布的压力如图 5-4 所示。在结点 i 的强度为 p_i,在结点 j 的强度为 p_j,ij边的长度为 l。利用面积坐标插值 ij边上任意一点处的压力为

$$p=p_iN_i+p_jN_j\tag{5-77}$$

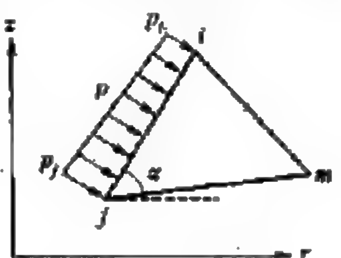


图 5-4　线性分布的压力

把它分解成沿 r 和 z 方向的两个分量为

$$\begin{Bmatrix}p_r\\p_z\end{Bmatrix}=\begin{Bmatrix}p\sin\alpha\\-p\cos\alpha\end{Bmatrix}=\begin{Bmatrix}p(z_i-z_j)/l\\p(r_j-r_i)/l\end{Bmatrix}\tag{5-78}$$

因此,在结点 i 上的等效结点力是

$$f_{si}^{e}=\begin{Bmatrix}f_{sri}\\f_{szi}\end{Bmatrix}=p_i\begin{Bmatrix}z_i-z_j\\r_j-r_i\end{Bmatrix}\tag{5-79}$$

式中

$$\bar{p}_i=\frac{\pi}{6}[p_i(3r_i+r_j)+p_j(r_i+r_j)]\tag{5-80}$$

同样可以得到结点 j 上的等效结点力是

$$f_{sj}^{e}=\begin{Bmatrix}f_{srj}\\f_{szj}\end{Bmatrix}=p_j\begin{Bmatrix}z_i-z_j\\r_j-r_i\end{Bmatrix}\tag{5-81}$$

式中

$$\bar{p}_j=\frac{\pi}{6}[p_i(r_i+r_j)+p_j(r_i+3r_j)]\tag{5-82}$$

这里给出了计算线性分布压力的等效结点力的 MATLAB 程序段。

```
1.   function enf = EquivalentDistPressure( m, n, pm, pn )
2.   % 计算线性分布压力的等效结点力
3.   % 输入参数:
4.   %   m,n ------------ 结点号
5.   %   pm,pn ---------- 与结点号对应的压力值
6.   % 返回值:
7.   %   enf ------------ 等效结点力向量
8.       global gNode
```

```
9.    enf = zeros(4,1);
10.
11.    % 获取结点坐标
12.    rm = gNode(m,1);
13.    rn = gNode(n,1);
14.    zm = gNode(m,2);
15.    zn = gNode(n,2);
16.
17.    % 计算 pm_,pn_
18.    pm_ = pi/6*(pm*(3*rm+rn)+pn*(rm+rn));
19.    pn_ = pi/6*(pm*(rm+rn)+pn*(rm+3*rn));
20.
21.    % 计算 3 个结点上的等效结点力
22.    enf(1:2) = pm_*[zm-zn;rn-rm];
23.    enf(3:4) = pn_*[zm-zn;rn-rm];
24.  return
```

另一种常见的分布表面力方向是固定不变的，此时公式比分布压力更为简单。不妨设线性分布表面力的方向为径向，如图 5-5 所示。在结点 i 的大小为 p_i，在结点 j 的大小为 p_j，ij 边的长度为 l。在此情况下有 $p_r = p_iN_i + p_jN_j, p_z = 0$，于是等效结点力为

$$\boldsymbol{f}_{si} = \begin{Bmatrix} f_{sri} \\ f_{szi} \end{Bmatrix} = 2\pi\int N_i \begin{Bmatrix} N_ip_i + N_jp_j \\ 0 \end{Bmatrix} r\mathrm{d}s \tag{5-83}$$

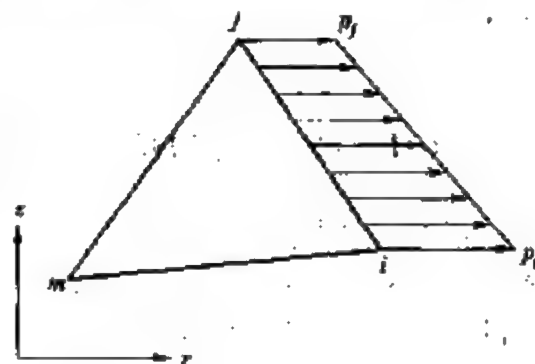


图 5-5 径向的分布力

利用面积坐标积分公式，得到

$$\boldsymbol{f}_{si} = \begin{Bmatrix} p_il \\ 0 \end{Bmatrix} \quad (i,j) \tag{5-84}$$

最后，如果考虑温度改变的影响，单元中温度改变引起的结点力为

$$\boldsymbol{f}_{Ti} = 2\pi\iint \boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{D}\boldsymbol{\varepsilon}_0 r\mathrm{d}r\mathrm{d}z \quad (i,j,m) \tag{5-85}$$

对于各向同性材料,温度应变为

$$\boldsymbol{\varepsilon}_0 = \alpha T \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix}^{\mathrm{T}} \tag{5-86}$$

把 $\boldsymbol{B}_i$ 和各向同性材料的弹性矩阵 $\boldsymbol{D}$ 代入式(5-85),得到

$$\boldsymbol{f}_{Ti}^{e} = \frac{\alpha\pi E}{\Delta(1-2\mu)} \iint \begin{Bmatrix} (b_i + g_i)T \\ c_i T \end{Bmatrix} r\mathrm{d}r\mathrm{d}z \qquad (i,j,m) \tag{5-87}$$

上式中的两个积分通常可用数值积分求得，或者简单地取近似表达式

$$\left.\begin{aligned} \iint (b_i + g_i) Tr\mathrm{d}r\mathrm{d}z &\approx (b_i + \bar{g}_i)\bar{r}\Delta\bar{T} \\ \iint c_i Tr\mathrm{d}r\mathrm{d}z &= c_i\bar{r}\Delta\bar{T} \end{aligned}\right\} ,\bar{T} = (T_i + T_j + T_m)/3 \tag{5-88}$$

最后得到

$$\boldsymbol{f}_{Ti}^{e} = \frac{\alpha\pi E\,\bar{r}\bar{T}}{(1-2\mu)} \begin{Bmatrix} b_i + \bar{g}_i \\ c_i \end{Bmatrix} \qquad (i,j,m) \tag{5-89}$$

5.6 算例

5.6.1 单向拉伸的杆

设有单向拉伸的杆,如图 5-6 所示,长 × 宽 × 高 = 10m × 2m × 2m,两端受 400×10^6N 的拉力。由于对称，取如图 5-6 所示梁的 1/8 进行计算。这里要注意的是，由于在拉力作用处 A 点,梁被分成了 4 等分，因此在有限元模型中 A 点的集中力也要取总拉力的 1/4,即 100×10^6N。同样,根据对称性,在 $y=0$ 的面上 y 方向的位移为零,在 $x=0$ 的面上 x 方向的位移为零,$z=0$ 的面上 z 方向的位移为零。

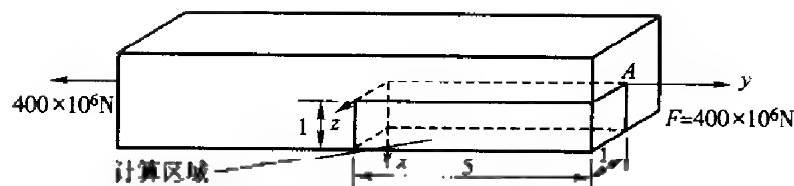


图 5-6　单向拉伸的杆

用四面体单元对计算区域进行离散，共1308个结点，5730个单元，模型如图5-7所示。图5-8显示了在离开集中力作用点*A*不同距离处，拉应力在截面上的分布情况。从图中可以明显看出，随着与*A*点的距离增加，截面上的应力分布迅速趋于均布。当距离*A*点2m处，也就是$y=3$m处，应力分布基本上已经是均匀分布。这从一个方面验证了圣维南原理。

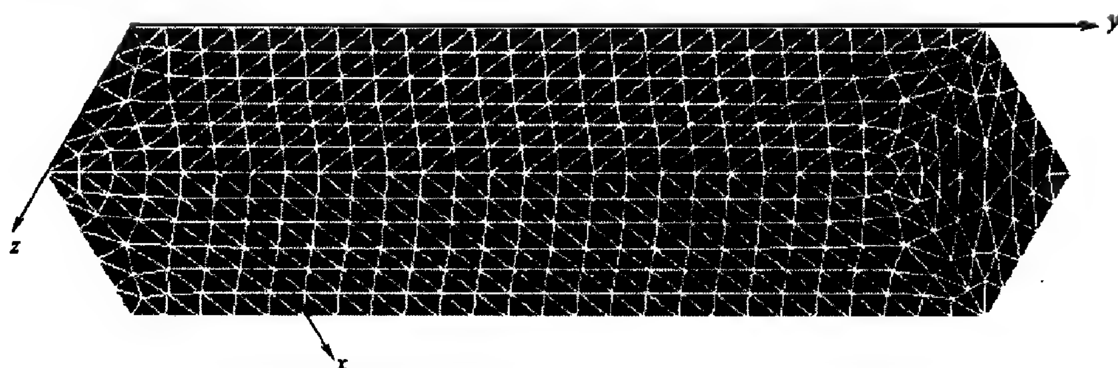


图5-7 单向拉伸杆的有限元网格

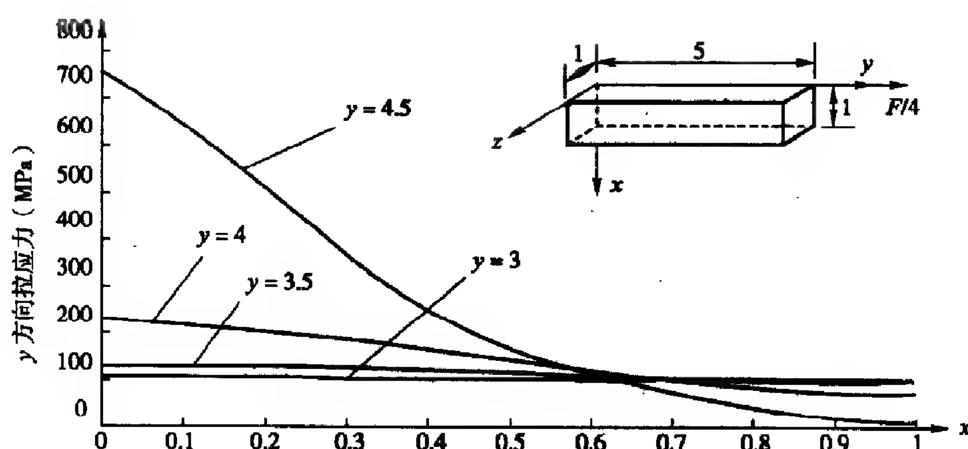


图5-8 在$z=0$平面上y方向拉应力沿x轴的变化

5.6.2 圆形垂直荷载作用下的弹性半空间

弹性半空间在圆形荷载作用下的变形和应力分析是弹性力学中的经典问题，在工程中也很有实用价值。有限元求解这类无限域或半无限域的问题时，求解区域取多大是关键。一般要根据问题的性质，并没有统一的解决方法。此时有限元知识、力学知识和工程经验就显得十分重要。利用**Boussinesq**给出的基本解，可以得到这个问题的解析解[12]。在分布荷载的中心点处的竖向位移是

$$w = 2qa(1-\nu^2)/E \tag{5-90}$$

式中，E、ν 是半空间的弹性模量和泊松比，q 是分布力，a 是分布力作用半径。本算例分析有限元的精度与所取区域大小的关系。

图 5-9 所示为有限元网格及边界条件。所取区域为深 H 宽 W 的矩形区域。为了避免重新划分网格的麻烦，这里采用固定 H 和 W，取不同的荷载作用半径 a 进行分析，这样可以消除不同网格布局带来的影响。表 5-1 给出了不同 a 时的有限元和解析解，并计算了相对误差，发现当 W/a 越来越大时，有限元的结果逐渐趋近于解析解。从本算例可以看出，当取分析区域为分布荷载作用半径的 10 倍时，得到的位移精度在 5% 左右，这可以满足一般的工程要求。这里需要注意的是，本算例只是针对均匀的弹性半空间做了研究，如果对于分层地基，那么各层弹性模量之间的差异对分析区域的范围要求有很大的影响。

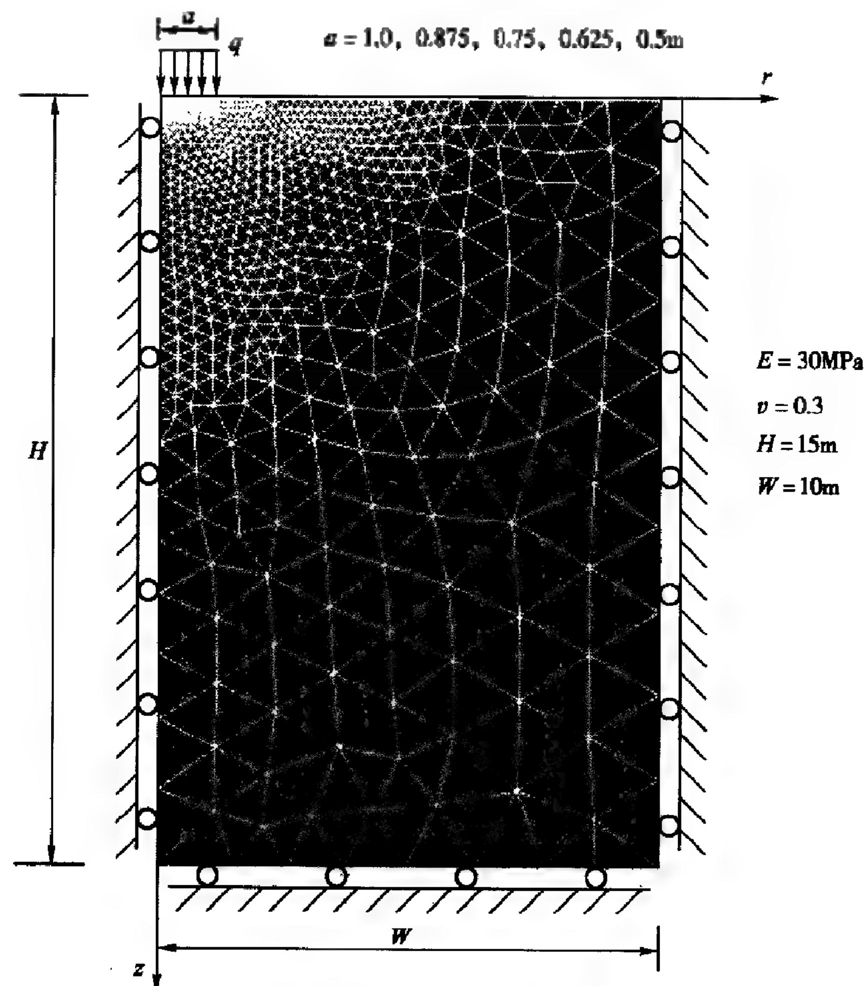


图 5-9　在圆形垂直荷载作用下半无限空间的有限元模型

不同荷载作用半径下荷载圆心处的竖向位移 w　　表 5-1

a	W/a	有限元(mm)	解析解[12](mm)	相对误差(%)
1.000	10.0	5.750	6.067	-5.225
0.875	11.4	5.066	5.308	-4.559
0.750	13.3	4.368	4.550	-4.000
0.625	16.0	3.660	3.792	-3.481
0.500	20.0	2.943	3.033	-2.967

另外，本算例是对于同一个有限元网格进行不同荷载下的计算，因此有限元模型的整体刚度矩阵只需要计算一次，另外求解最后的线性方程组时也只要进行一次三角分解。而这两个步骤在有限元程序中是用时最多的，因此组织合理的计算步骤对提高有限元的计算速度是十分重要的。在 MATLAB 中，对于 m 个具有相同系数矩阵 $\boldsymbol{A}$，不同右端项 $\boldsymbol{b}_1,\boldsymbol{b}_2,\cdots,\boldsymbol{b}_m$ 的线性方程组

$$\boldsymbol{AX}=\boldsymbol{B} \tag{5-91}$$

式中

$$\boldsymbol{X}=[\boldsymbol{x}_1\quad \boldsymbol{x}_2\quad \cdots\quad \boldsymbol{x}_m],\quad \boldsymbol{B}=[\boldsymbol{b}_1\quad \boldsymbol{b}_2\quad \cdots\quad \boldsymbol{b}_m] \tag{5-92}$$

可以一次性求解。具体应用时，对不同的荷载生成相应的右端项，然后一次求解。具体的程序段如下：

```
1.  %  计算单元刚度矩阵,并集成到整体刚度矩阵中
2.  [element_number,dummy] = size(gElement);
3.  for ie=1:1:element_number
4.      disp(sprintf( '计算刚度矩阵,当前单元: %d', ie ));
5.      k = StiffnessMatrix( ie );
6.      AssembleStiffnessMatrix( ie, k );
7.  end
8.  %  计算分布压力产生的等效结点力
9.  for idfstep = 1:length(gDFStep)
10.     df_number = gDFStep( idfstep );
11.     for i =1 + sum(gDFStep(1:idfstep-1)):df_number +
                              sum(gDFStep(1:idfstep-1))
12.     enf = EquivalentDistPressure( gDF(i,1),
                   gDF(i,2), gDF(i,3), gDF(i,4) );
```

```
13.     m = gDF(i, 1) ;
14.     n = gDF(i, 2) ;
15.     f(m* 2 -1:m* 2,idfstep) = f(m* 2 -1:m* 2,idfstep) +enf(1:2) ;
16.     f(n* 2 -1:n* 2,idfstep) = f(n* 2 -1:n* 2,idfstep) +enf(3:4) ;
17.   end
18.   end
19.   %   处理约束条件,修改刚度矩阵和结点力向量。采用划行划列法
20.   [bc_number,dummy] = size(gBC1) ;
21.   for ibc =1:1:bc_number
22.       n = gBC1(ibc, 1) ;
23.       d = gBC1(ibc, 2) ;
24.       m = (n - 1)*2 + d ;
25.       for idfstep = 1:length(gDFStep)
26.       f(:,idfstep) = f(:,idfstep) - gBC1(ibc,3) * gK(:,m) ;
27.       f(m,idfstep) = gBC1(ibc,3) ;
28.       end
29.       gK(:,m) = zeros(node_number* 2, 1) ;
30.       gK(m,:) = zeros(1, node_number* 2) ;
31.       gK(m,m) = 1.0 ;
32.   end
33.   %   一次性求解方程组,得到结点位移向量
34.   gDelta = gK \ f ;
```

这里变量 gDFStep 保存了分布荷载的种类个数，变量 f 是右端的整体结点力，每一列对应一种分布荷载。在集成整体结点力和处理边界条件，对应每一种分布荷载的每一列都进行对应的处理，最后就可以一次性求解。从程序中可以看出，整体刚度矩阵 gK 只形成一次，方程组求解也只进行一次。

参考文献

[1] Timoshenko S. P. Goodier J. N.. Theory of elasticity. New York: McGraw-Hill, 1970.

[2] 徐芝纶. 弹性力学. 北京:高等教育出版社, 1986.

[3] Martin H. C.. Plane elasticity problems and the direct stiffness method. the Trend in Engineering, 1961, 13: 5-19.

[4] Gallagher R. H., Padlog. J., Bijlaard P. P.. Stress analysis of heated complex shapes. Journal of the American Rocket Society, 1962: 700-707, May.

[5] 方德植.解析几何.北京:高等教育出版社. 1986.

[6] 王凯. N层弹性连续体系在圆形均布垂直荷载作用下的力学计算. 土木工程学报, 1982, 15(2): 65-76.

[7] 钟阳, 王哲人, 郭大智. 求解多层弹性半空间轴对称问题的传递矩阵法. 土木工程学报, 1992, 25(6): 37-43.

[8] Clough R., Rashid Y.. Finite element analysis of axisymmetric solids. Journal of the Engineering Mechanics Division, ASCE, 1965, 91: 71-85.

[9] Wilson E.. Structural analysis of axisymmetric solids, Journal of the American Institute of Aeronautics and Astronautics, 1965, 3(12): 2269-2274.

[10] Utku. S.. Explicit expression for triangular torus element stiffness matrix. Journal of the American Institute of Aeronautics and Astronautics, 1968, 6(6): 1174-1176.

[11] 丁皓江, 何福保, 谢贻权, 徐兴. 弹性和塑性力学中的有限单元法. 2版. 北京: 机械工业出版社, 1989.

[12] Davis R. O., Selvadurai A. P. S.. Elasticity and Geomechanics. Cambridge University Press, 1996.

第6章 等参数单元

前面几章中给出的单元都是建立在物理空间下的，主要目的是能容易地理解有限元法。但是，在实际应用中，等参数单元更加有效[1~5]。等参数单元的形函数一般建立在一个参考坐标系下，它既被用来进行单元内位移插值，也被用来表示单元内任意一点的坐标。经过形函数的坐标变换，可以把物理空间下任意的四边形或六面体，甚至是曲边的四边形或曲面的六面体转换成参考坐标系下的正方形或立方体。这些曲边或曲面的单元可以更精确地描述求解区域，而且应用更高阶的位移插值函数，因此具有较高的精度。另外，等参数单元形函数选择方法的统一性，非常有利于程序实现。

6.1 平面等参数单元

在平面问题中，我们曾经介绍了两种最简单的单元，三角形单元和矩形单元。所采用的是线性和双线性的位移模式，它们是对实际位移分布的最低级逼近，精度有限。矩形单元难以应用于非规则边界和构造梯度网格以获取关键区域的细节。这节介绍的平面等参数单元是直边或曲边的任意四边形，可以适应不规则的边界和构造梯度网格，并具有较高的精度。

6.1.1 四结点单元

图6-1a）是一个物理空间中的任意四边形单元。在它上面建立一个参考坐标系$\xi\eta$。在物理空间中，它们不需要相互正交，也

不需要与整体坐标系平行。单元的边被 ξ 和 η 平分，它们的方程分别是 $\xi=\pm1$ 和 $\eta=\pm1$，如图 6-1a)所示。在参考坐标系 $\xi\eta$ 下，该四边形变换成一个边长为 2 的正方形，如图 6-1b)所示。

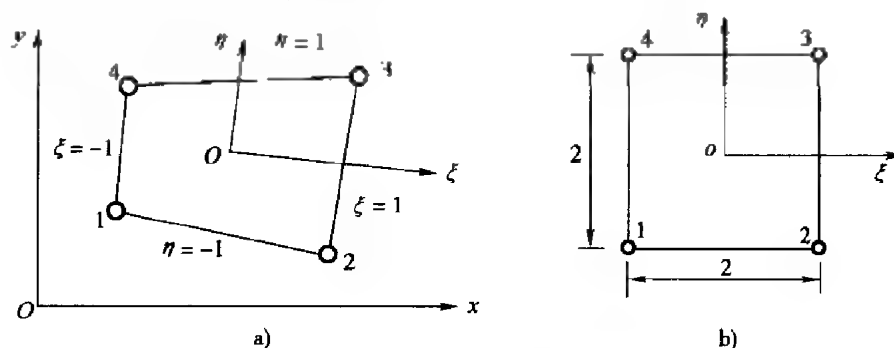


图 6-1　四结点四边形等参数单元

根据等参数单元的含义，它的位移和坐标都采用相同的形函数表示，即

$$u=\sum_{i=1}^{4}N_i(\xi,\eta)u_i,\quad v=\sum_{i=1}^{4}N_i(\xi,\eta)v_i \tag{6-1}$$

$$x=\sum_{i=1}^{4}N_i(\xi,\eta)x_i,\quad y=\sum_{i=1}^{4}N_i(\xi,\eta)y_i \tag{6-2}$$

式中 u_i 和 $v_i(i=1,2,3,4)$ 是整体坐标系 xy 下结点的位移分量，x_i 和 y_i 是结点的整体坐标，$N_i(\xi,\eta)$ 是由参考坐标 ξ 和 η 表示的形函数。

根据第 4 章中矩形单元的形函数，N_i 的表达式可写成

$$N_i=\frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta)\qquad(i=1,2,3,4) \tag{6-3}$$

式中，ξ_i 和 η_i 是 4 个结点的参考坐标值，它们的值为 ±1。这里需要注意的是，形函数由式(6-3)给定时，参考坐标 ξ 和 η 在整体坐标系 xy 下的方向是由单元的结点编号决定的。图 6-1 所示的单元中，结点编号是 1-2-3-4。如果把结点顺序换成 4-1-2-3，那么图 6-1a)中的 ξ 和 η 要绕原点顺时针旋转，使 η 轴和原来的 ξ 轴重合，而 ξ 轴指向原来 η 轴的负方向。

6.1.2　单元的特性分析

将位移的表达式(6-1)代入几何方程，便得到应变分量的计算公式

$$\boldsymbol{\varepsilon}=\begin{Bmatrix}\dfrac{\partial u}{\partial x}\\[2ex]\dfrac{\partial v}{\partial y}\\[2ex]\dfrac{\partial u}{\partial y}+\dfrac{\partial v}{\partial x}\end{Bmatrix}=\boldsymbol{B}\boldsymbol{\delta}^e=[\boldsymbol{B}_1\quad\boldsymbol{B}_2\quad\boldsymbol{B}_3\quad\boldsymbol{B}_4]\boldsymbol{\delta}^e \tag{6-4}$$

式中 $\boldsymbol{\delta}^e=[\boldsymbol{\delta}_1^T \quad \boldsymbol{\delta}_2^T \quad \boldsymbol{\delta}_3^T \quad \boldsymbol{\delta}_4^T]^T$是单元结点的位移列阵，以及

$$\boldsymbol{B}_i=\begin{bmatrix} N_{i,x} & 0 \\ 0 & N_{i,y} \\ N_{i,y} & N_{i,x} \end{bmatrix},\quad \boldsymbol{\delta}_i=\begin{Bmatrix} u_i \\ v_i \end{Bmatrix}\quad (i=1,2,3,4) \tag{6-5}$$

这里下标中的逗号表示求导数，即 $N_{i,x}=\partial N_i/\partial x, N_{i,y}=\partial N_i/\partial y$。由于形函数是由参考坐标 ξ 和 η 给出的，这两个导数一般不能显式给出。根据复合函数求导规则，它们可以按下列步骤计算

$$\begin{Bmatrix} N_{i,\xi} \\ N_{i,\eta} \end{Bmatrix}=\begin{bmatrix} x_{,\xi} & y_{,\xi} \\ x_{,\eta} & y_{,\eta} \end{bmatrix}\begin{Bmatrix} N_{i,x} \\ N_{i,y} \end{Bmatrix}=\boldsymbol{J}\begin{Bmatrix} N_{i,x} \\ N_{i,y} \end{Bmatrix} \tag{6-6}$$

式中的矩阵 $\boldsymbol{J}$ 称为雅可比(Jacobi)矩阵，即

$$\boldsymbol{J}=\begin{bmatrix} x_{,\xi} & y_{,\xi} \\ x_{,\eta} & y_{,\eta} \end{bmatrix}=\begin{bmatrix} \sum_{i=1}^{4} N_{i,\xi}x_i & \sum_{i=1}^{4} N_{i,\xi}y_i \\ \sum_{i=1}^{4} N_{i,\eta}x_i & \sum_{i=1}^{4} N_{i,\eta}y_i \end{bmatrix} \tag{6-7}$$

则式(6-5)中形函数对 x 和 y 的导数为

$$\begin{Bmatrix} N_{i,x} \\ N_{i,y} \end{Bmatrix}=\boldsymbol{J}^{-1}\begin{Bmatrix} N_{i,\xi} \\ N_{i,\eta} \end{Bmatrix} \tag{6-8}$$

单元内的应力根据本构关系，可以表示成

$$\boldsymbol{\sigma}=\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}=\boldsymbol{D\varepsilon}=\boldsymbol{DB\delta}^e=\boldsymbol{S\delta}^e \tag{6-9}$$

式中的应力矩阵 $\boldsymbol{S}$ 也可以写成分块的形式

$$\boldsymbol{S}=[\boldsymbol{S}_1 \quad \boldsymbol{S}_2 \quad \boldsymbol{S}_3 \quad \boldsymbol{S}_4] \tag{6-10}$$

对于平面应力情形，应变子矩阵为

$$\boldsymbol{S}_i=\boldsymbol{DB}_i=\frac{E}{1-\mu^2}\begin{bmatrix} N_{i,x} & \mu N_{i,y} \\ \mu N_{i,x} & N_{i,y} \\ (1-\mu)N_{i,y}/2 & (1-\mu)N_{i,x}/2 \end{bmatrix} \tag{6-11}$$

对于平面应变，则只需要把 E 换成 $E/(1-\mu^2)$，把 μ 换成 $\mu/(1-\mu)$即可。根据单元刚度的普遍公式(2-70)，四结点四边形等参数单元的刚度矩阵是

$$\boldsymbol{K}^e=\iint_{\Omega}\boldsymbol{B}^T\boldsymbol{DB}h\mathrm{d}x\mathrm{d}y \tag{6-12}$$

式中 h 是单元的厚度，单元刚度矩阵可划分成 16 块 2×2 的子矩阵的形式。另外，由于积分中的应变矩阵 $\boldsymbol{B}$ 是由参考坐标 ξ 和 η 表示的，因此对积分区域进行相应的积分变换，得到子矩阵的计算公式是

$$\boldsymbol{K}_{ij} = \iint_{\Omega} \boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}_j\, h\mathrm{d}x\mathrm{d}y = \int_{-1}^{1}\int_{-1}^{1} \boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}_j\,|\boldsymbol{J}|\,h\mathrm{d}\xi\mathrm{d}\eta \qquad (6\text{-}13)$$

$$(i,j = 1,2,3,4)$$

对于平面应力情形

$$\boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}_j = \frac{E}{1-\mu^2}\begin{bmatrix} N_{i,x}N_{j,x} + \dfrac{1-\mu}{2}N_{i,y}N_{j,y} & \mu N_{i,x}N_{j,y} + \dfrac{1-\mu}{2}N_{i,y}N_{j,x} \\ \mu N_{i,y}N_{j,x} + \dfrac{1-\mu}{2}N_{i,x}N_{j,y} & N_{i,y}N_{j,y} + \dfrac{1-\mu}{2}N_{i,x}N_{j,x} \end{bmatrix} \qquad (6\text{-}14)$$

$$(i,j=1,2,3,4)$$

由于形函数对整体坐标 x 和 y 的导数中包含雅可比矩阵的逆矩阵，式(6-13)的积分一般情况下不能得到显式。因此要采用后面介绍的数值积分来求得等参数单元的刚度矩阵。

6.1.3 八结点四边形单元

前面给出的四结点单元虽然在几何形状上比矩形单元自由得多，但是位移模式仍然是双线性的，其计算精度跟矩形单元是一样的。如果在四边形的边上增加结点，则可以构造更复杂适应性更强的单元，因为四边形的边甚至可以是弯曲的，如图 6-2 所示。

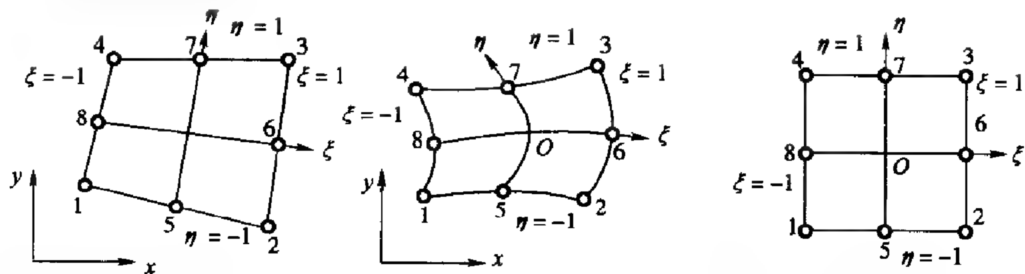


图 6-2 八结点四边形单元

曲边单元能更好地模拟那些具有弯曲边界的结构。进行坐标变换后，参考坐标系 $\xi\eta$ 下仍然是一个边长为 2 的正方形，而且边上的结点 5、6、7 和 8 被映射到正方形 4 个边的中点。形函数的确定有一个方便的方法。以结点 5 为例，在参考坐标系下的正方形单元中，将不含结点 5 的 3 个边的方程相乘，并使这个乘积在结点 5 有等于 1 的函数值，即

$$N_5=(1-\xi^2)(1-\eta)/2 \tag{6-15}$$

其他几个边结点的形函数类似地有

$$N_6=(1+\xi)(1-\eta^2)/2 \tag{6-16}$$

$$N_7=(1-\xi^2)(1+\eta)/2 \tag{6-17}$$

$$N_8=(1-\xi)(1-\eta^2)/2 \tag{6-18}$$

接着来看角结点 1、2、3 和 4 的形函数。四结点单元的形函数 N_1 在结点 5 和 8 处的值为 0.5，因此对于八结点单元，$\frac{1}{4}(1-\xi)(1-\eta)-\frac{1}{2}(N_5+N_8)$ 在结点 1 的值为 1，在其余结点处的值为零，所以它就是八结点单元的形函数 N_1，即

$$N_1=\frac{1}{4}(1-\xi)(1-\eta)-\frac{1}{2}(N_5+N_8) \tag{6-19}$$

其他几个角结点有类似的形函数

$$N_2=\frac{1}{4}(1+\xi)(1-\eta)-\frac{1}{2}(N_5+N_6) \tag{6-20}$$

$$N_3=\frac{1}{4}(1+\xi)(1+\eta)-\frac{1}{2}(N_6+N_7) \tag{6-21}$$

$$N_4=\frac{1}{4}(1-\xi)(1+\eta)-\frac{1}{2}(N_7+N_8) \tag{6-22}$$

如果增加一个约定，即某一边上结点不出现，就令其对应的形函数为零，那么式(6-15)～式(6-22)就是 4～8 结点单元的形函数。有了这个约定，只要把前面四结点单元的所有涉及形函数求和的范围从 $i=1,2,3,4$ 换成 $i=1,2,\cdots,n$ $(n=4\sim8)$就是 4～8 结点单元相应的公式。这样，前面推导的公式全部适用 4～8 结点单元。如果在单元的每条边上增加两个结点，就可构造 12 结点等参数单元。其形函数也可按上述方法得到，具体可参考本章参考文献[5]

6.1.4 高斯积分法和单元刚度矩阵的计算

高斯(Gauss)积分法[6]是最常用的一种数值积分方法。它把积分化成采样点函数值的加权和。例如，对于一维积分，有

$$\int_{-1}^{1} f(\xi)\mathrm{d}\xi = \sum_{i=1}^{n} W_i f(\xi_i) \tag{6-23}$$

式中，ξ_i 和 W_i 分别是采样点和相应的权系数，n 是采样点的个数，称为积分的阶数。

权系数的确定准则是取被积函数为多项式，并使其误差最小，因此在相同的精度下，它比其它的数值使用更少的采样点。一般用术语“高斯点”来表示

这些积分点，它们对称于积分区域的中心成对出现，是 **Legendre** 多项式的根。对称的高斯点处有相同的权系数。表 **6-1** 给出了积分阶数分别是 **1**、**2** 和 **3** 时的高斯点和权系数。更高阶的可参考本节参考文献[6]，但是在有限元中并不常用。

高斯求积法中的积分点坐标和加权系数　　　　表 6-1

积分阶数	代数精度	高斯点	权系数
1	1	0	2
2	3	$\pm 1/\sqrt{3} = \pm 0.577350269189626$	1
3	5	$\pm\sqrt{0.6} = \pm 0.774596669241483$ 0	$5/9 = 0.555555555555555$ $8/9 = 0.888888888888888$

设有二阶的高斯积分公式

$$\int_{-1}^{1} f(\xi)\mathrm{d}\xi = W_1 f(\xi_1) + W_2 f(\xi_2) \tag{6-24}$$

其中高斯点 $\xi_{1,2} = \mp 1/\sqrt{3}$ 和权系数 $W_{1,2} = 1$，下面来验证它具有的积分精度。分别设 $f(\xi)$ 为 1、ξ、ξ^2、ξ^3 和 ξ^4，得到

$f(\xi)$	$\int_{-1}^{1} f(\xi)\mathrm{d}\xi$		$W_1 f(\xi_1) + W_2 f(\xi_2)$
1	2	=	2
ξ	0	=	0
ξ^2	2/3	=	2/3
ξ^3	0	=	0
ξ^4	2/5	≠	2/9

由此可以看出，两点的高斯积分公式对于三次多项式都精确成立。其实可以证明，n 阶高斯积分公式对于 $2n-1$ 次多项式都精确满足[6]。

对不同的积分变量顺序使用一维的高斯积分公式就能得到多维情形的高斯积分公式，如

$$\begin{aligned}\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)\mathrm{d}\xi\mathrm{d}\eta &= \int_{-1}^{1}\Big[\sum_{i=1}^{m} W_i f(\xi_i,\eta)\Big]\mathrm{d}\eta \\ &= \sum_{j=1}^{n} W_j\Big[\sum_{i=1}^{m} W_i f(\xi_i,\eta_j)\Big] = \sum_{i=1}^{m}\sum_{j=1}^{n} W_i W_j f(\xi_i,\eta_j)\end{aligned} \tag{6-25}$$

同样三维的高斯积分公式为

$$\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta,\zeta)\,\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta = \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{l} W_i W_j W_k f(\xi_i,\eta_j,\zeta_k) \tag{6-26}$$

根据高斯积分法，可以写出四结点至八结点等参数单元刚度矩阵数值积分公式为

$$\boldsymbol{K}^e = h\sum_{i=1}^{m}\sum_{j=1}^{n} W_i W_j (\boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}|\boldsymbol{J}|)_{ij} \tag{6-27}$$

式中,圆括号的下标 ij 表示圆括号内的部分取在高斯点 (ξ_i,η_j) 处的值。在具体程序中一般采用两重循环来实现两个求和，即

初始化单元刚度 $\boldsymbol{K}^e$ 并清零

┌在 ξ 方向的积分点上循环($i=1:m$)
│┌在 η 方向的积分点上循环($j=1:n$)
││　计算高斯点(ξ_i,η_j)处 $\boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}|\boldsymbol{J}|=\boldsymbol{F}(\xi_i,\eta_j)$的值
││　$\boldsymbol{K}^e=\boldsymbol{K}^e+W_iW_j\boldsymbol{F}(\xi_i,\eta_j)$
│└j 循环结束
└i 循环结束

其中计算 $\mathbf{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}|\boldsymbol{J}|=\boldsymbol{F}(\xi_i,\eta_j)$ 的值是最花时间的操作。因为要计算形函数的导数，Jacobi 矩阵的逆矩阵及其行列式值，并进行矩阵相乘。为了增加程序的可读性和结构化，一般把这些计算写成几个子程序供调用。

还有一个问题是具体应用时如何确定高斯积分的阶数。因为阶数不能取得过大，否则计算工作量将会随着积分点数的增加而急剧地增加。一般情况下，在平面问题中，物体划分成较多个等参数单元时，使用 2×2 的高斯积分法则，通常能取得良好得结果。其原因是，位移插值函数中的不完全的高次项，往往对计算精度起不良影响，降阶积分的措施正好相当于把那些不完全的高次项对刚度矩阵的影响去除掉[5]。

下面是利用高斯积分法计算平面等参数单元的刚度矩阵函数以及它所调用的一些函数。其中主要有计算应变矩阵 $\boldsymbol{B}$ 的函数，计算 Jacobi 矩阵的函数，计算形函数对整体坐标和局部坐标导数的函数。注意这里计算的是单位厚度的刚度矩阵，因此没有出现厚度 h。

```
1 [illegible]
2 % 计算平面[illegible]刚度矩阵
3 [illegible] 输入参数:
```

```
4.  %      ie -- [illegible]
5.  %  [illegible]:
6.  %      k  -- [illegible]
7.  %  [illegible]:
8.  %      [illegible]
9.  %
10.     global gElement gMaterial
11.     k = zeros( 16, 16 ) ;
12.     E  = gMaterial( gElement(ie, 9), 1 ) ;   % [illegible]
13.     mu = gMaterial( gElement(ie, 9), 2 ) ;   % [illegible]
14.     % [illegible]
15.     A1 = mu ;
16.     A2 = (1-mu)/2 ;
17.     A3 = E/(1-mu*mu) ;
18.     % [illegible]
19.     % A1 = mu/(1-mu) ;
20.     % A2 = (1-2*mu)/2/(1-mu) ;
21.     % A3 = E*(1-mu)/(1+mu)/(1-2*mu) ;
22.     D = A3* [ 1  A1  0
23.               A1  1  0
24.               0   0  A2] ;
25.     % 3 x 3 [illegible]
26.     % x = [ -0.774596669241483, 0.0, 0.774596669241483 ] ;
27.     % w = [ 0.555555555555556, 0.888888888888889, 0.555555555555556 ] ;
28.     % 2 x 2 [illegible]
29.     x = [ -0.577350269189626, 0.577350269189626 ] ;
30.     w = [1, 1] ;
31.     for i=1:1:length(x)
32.         for j=1:1:length(x)
33.             B = MatrixB( ie, x(i), x(j) ) ;
34.             J = Jacobi( ie, x(i), x(j) ) ;
35.             k = k + w(i)*w(j)*transpose(B)*D*B*det(J) ;
36.         end
37.     end
38. return
39.
```

```
40.  function B = MatrixB( ie, xi, eta )
41.  %  计算单元的应变矩阵B
42.  %  输入参数:
43.  %     ie ---------- 单元号
44.  %     xi,eta ------ 局部坐标
45.  %  返回值:
46.  %     B  ---------- 在局部坐标处的应变矩阵B
47.
48.    [N_x,N_y] = N_xy( ie, xi, eta );
49.    B = zeros( 3, 16 );
50.    for i=1:1:8
51.        B(1:3,(2*i-1):2*i) = [ N_x(i)      0
52.                                0      N_y(i)
53.                              N_y(i)  N_x(i)];
54.    end
55.  return
56.
57.  function [N_x, N_y] = N_xy( ie, xi, eta )
58.  %  计算形函数对整体坐标的导数
59.  %  输入参数:
60.  %     ie ---------- 单元号
61.  %     xi,eta ------ 局部坐标
62.  %  返回值:
63.  %     N_x --------- 在局部坐标处的形函数对x坐标的导数
64.  %     N_y --------- 在局部坐标处的形函数对y坐标的导数
65.
66.    J = Jacobi( ie, xi, eta );
67.    [N_xi,N_eta] = N_xieta( ie, xi, eta );
68.    A=inv(J)*[N_xi;N_eta];
69.    N_x = A(1,:);
70.    N_y = A(2,:);
71.  return
72.
73.  function [N_xi, N_eta] = N_xieta( ie, xi, eta )
74.  %  计算形函数对局部坐标的导数
75.  %  输入参数:
```

```
76.  %      ie ----------- 单元号
77.  %      xi,eta -------- 局部坐标
78.  %  返回值:
79.  %      N_xi ---------- [illegible]
80.  %      N_eta --------- [illegible]
81.
82.    x = [ -1, 1, 1, -1 ] ;
83.    e = [ -1, -1, 1, 1 ] ;
84.    N_xi  = zeros( 1, 8 ) ;
85.    N_eta = zeros( 1, 8 ) ;
86.
87.    N_xi( 5 )  = xi* (eta - 1) ;
88.    N_eta( 5 ) = 0.5* (xi^2 - 1) ;
89.    N_xi( 6 )  = 0.5* (1 - eta^2) ;
90.    N_eta( 6 ) = - eta* (xi + 1) ;
91.    N_xi( 7 )  = - xi* (eta + 1) ;
92.    N_eta( 7 ) = 0.5* (1 - xi^2) ;
93.    N_xi( 8 )  = 0.5* (eta^2 - 1) ;
94.    N_eta( 8 ) = eta* (xi - 1) ;
95.
96.    N_xi(1)  = x(1)* (1 + e(1)* eta)/4...
                 - 0.5* ( N_xi(5)  + N_xi(8) ) ;
97.    N_eta(1) = e(1)* (1 + x(1)* xi)/4...
                 - 0.5* ( N_eta(5) + N_eta(8) ) ;
98.    N_xi(2)  = x(2)* (1 + e(2)* eta)/4...
                 - 0.5* ( N_xi(5)  + N_xi(6) ) ;
99.    N_eta(2) = e(2)* (1 + x(2)* xi)/4...
                 - 0.5* ( N_eta(5) + N_eta(6) ) ;
100.   N_xi(3)  = x(3)* (1 + e(3)* eta)/4...
                 - 0.5* ( N_xi(6)  + N_xi(7) ) ;
101.   N_eta(3) = e(3)* (1 + x(3)* xi)/4...
                 - 0.5* ( N_eta(6) + N_eta(7) ) ;
102.   N_xi(4)  = x(4)* (1 + e(4)* eta)/4...
                 - 0.5* ( N_xi(7)  + N_xi(8) ) ;
103.   N_eta(4) = e(4)* (1 + x(4)* xi)/4...
                 - 0.5* ( N_eta(7) + N_eta(8) ) ;
```

```
104.  return
105.  [illegible]
106.  function J = Jacobi( ie, xi, eta )
107.  % [illegible]
108.  % [illegible]
109.  %     ie -------- 单元号
110.  %     xi,eta ---- 局部坐标
111.  %  返回值
112.  %     J --------- 在局部坐标(xi,eta)处的雅可比矩阵
113.      global gNode gElement
114.      x = gNode( gElement( ie, 1:8 ), 1 ) ;
115.      y = gNode( gElement( ie, 1:8 ), 2 ) ;
116.      [N_xi,N_eta] = N_xieta( ie, xi, eta ) ;
117.      x_xi  = N_xi  * x ;
118.      x_eta = N_eta * x ;
119.      y_xi  = N_xi  * y ;
120.      y_eta = N_eta * y ;
121.      J = [ x_xi, y_xi; x_eta, y_eta ];
122.  return
```

6.1.5 等效结点力计算

1. 集中力

通常将集中荷载作用点取为结点，此集中荷载作为这一结点的等效结点力。

2. 体积力

设单元的单位体积力是 $\boldsymbol{p}_v = [p_{vx} \quad p_{vy}]^{\mathrm{T}}$，则分配到单元各结点上的等效结点力按下式计算

$$\boldsymbol{f}_{vi} = \begin{Bmatrix} f_{vxi} \\ f_{vyi} \end{Bmatrix} = \int_{-1}^{1}\int_{-1}^{1} N_i \begin{Bmatrix} p_{vx} \\ p_{vy} \end{Bmatrix} h\,|\boldsymbol{J}|\,\mathrm{d}\xi\,\mathrm{d}\eta \quad (i = 1,2,\cdots,n) \tag{6-28}$$

积分可以利用高斯积分公式进行。如果体积力为简单函数，比如是多项式，那么上式可以得到显式的积分公式，但是十分繁琐。当然可以利用 **MATLAB** 的符号运算功能，得到结点上的等效结点力。但是利用高斯积分法更方便。

下面给出了计算旋转物体离心力的等效节点力程序段，可以看出程序十分简短，注意这里的密度 **ro** 是单位面积上的密度，因此计算时厚度 **h** 没有出现。

```
1.  function evf = EquivalentVolumeForce( ie )
2.  %  计算离心力的等效节点力
3.  %  输入参数:
4.  %     ie ---------- 单元号
5.  %  返回值:
6.  %     evf ---------- 等效节点力向量
7.      global gNode gElement gMaterial
8.
9.      evf = zeros( 16, 1 ) ;
10.     omega = 2094.0 ;    % 旋转角速度
11.     ro = gMaterial( gElement( ie, 9 ), 3 ) ;
12.     x = gNode( gElement( ie, 1:8 ), 1 ) ;
13.     y = gNode( gElement( ie, 1:8 ), 2 ) ;
14.     xi = [ -0.774596669241483,         0.0,0.774596669241483 ] ;
15.     w = [ 0.555555555555556,0.888888888888889,0.555555555555556 ] ;
16.     for i =1:length(xi)
17.         for j =1:length(xi)
18.             J = Jacobi( ie, xi(i), xi(j) ) ;
19.             detJ = det(J);
20.             N = ShapeFunction( xi(i), xi(j) ) ;
21.             evf(1:2:15) = evf(1:2:15) + N'*N*x*detJ*w(i)*w(j) ;
22.             evf(2:2:16) = evf(2:2:16) + N'*N*y*detJ*w(i)*w(j) ;
23.         end
24.     end
25.     evf = evf * ro * omega^2 ;
26. return
27.
28. function N = ShapeFunction( xi, eta )
29. %  计算形函数的值
30. %  输入参数:
31. %     ie ---------- 单元号
32. %     xi, eta ----- 单元内局部坐标
33. %  返回值:
```

```
34.  %    N ———————————— 形函数的值
35.  [illegible]
36.      N5 = ( eta - 1 ) * ( xi^2 - 1 ) / 2 ;
37.      N6 = ( xi + 1 ) * ( 1 - eta^2 ) / 2 ;
38.      N7 = ( eta + 1 ) * ( 1 - xi^2 ) / 2 ;
39.      N8 = ( xi - 1 ) * ( eta^2 - 1 ) / 2 ;
40.      N1 = ( 1 - xi ) * ( 1 - eta ) / 4 - 0.5 * ( N8 + N5 ) ;
41.      N2 = ( 1 + xi ) * ( 1 - eta ) / 4 - 0.5 * ( N5 + N6 ) ;
42.      N3 = ( 1 + xi ) * ( 1 + eta ) / 4 - 0.5 * ( N6 + N7 ) ;
43.      N4 = ( 1 - xi ) * ( 1 + eta ) / 4 - 0.5 * ( N7 + N8 ) ;
44.      N = [ N1 N2 N3 N4 N5 N6 N7 N8 ] ;
45.  return
```

3. 表面力

设单元的某边 Γ 上承受的单位表面力是 $\boldsymbol{p}_s=[p_{sx}\quad p_{sy}]^{\mathrm{T}}$，则在这条边上两个或 3 个结点的等效结点力分别按下式计算

$$f_{si}^e=\begin{Bmatrix}f_{sxi}\\f_{syi}\end{Bmatrix}=\int_\Gamma N_i\begin{Bmatrix}p_{sx}\\p_{sy}\end{Bmatrix}h\mathrm{d}s\quad（i\text{ 取 }\Gamma\text{ 边上的结点}）\tag{6-29}$$

而其它结点的形函数在该边上的值都为零，因此相应的等效结点力也为零。例如在如图 6-3 所示中 $\xi=1$ 的边 263 上作用有 x 方向的分布荷载，在结点 2、6 和 3 的值分别是 p_2、p_6 和 p_3。假设表面力的分布跟位移采用相同的插值，那么这条边上任意一点的分布力为

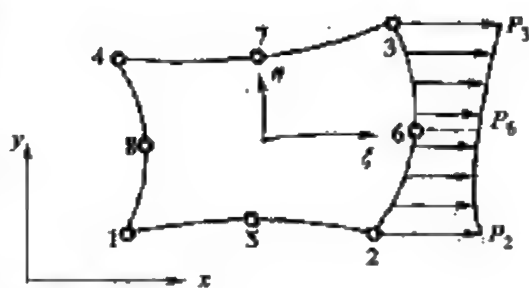


图 6-3　受表面力的四边形等参元

$$p_{sx}=N_2p_2+N_3p_3+N_6p_6\tag{6-30}$$

式中形函数 N_2、N_3 和 N_6 中取 $\xi=1$，即

$$N_2=-\frac{1}{2}\eta(1-\eta),\quad N_3=\frac{1}{2}\eta(1+\eta),\quad N_6=1-\eta^2\tag{6-31}$$

因此结点 2,3 和 6 上的等效结点力为

$$f_{xi}=\int_{-1}^{1}N_i(N_2p_2+N_3p_3+N_6p_6)\sqrt{(x_{,\eta})^2+(y_{,\eta})^2}\mathrm{d}\eta\quad(i=2,3,6)\tag{6-32}$$

式中

$$x_{,\eta}=N_{2,\eta}x_2+N_{6,\eta}x_6+N_{3,\eta}x_3, y_{,\eta}=N_{2,\eta}y_2+N_{6,\eta}y_6+N_{3,\eta}y_3 \tag{6-33}$$

式(6-32)中的积分一般不能得到解析式，可采用高斯积分公式进行数值积分。

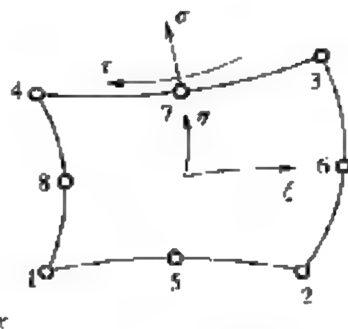


图 6-4 受法向和切向表面力的四边形等参元

如果给出的表面力是沿曲边的法向和切向力 $\boldsymbol{p}_t=[\sigma \quad \tau]^{\mathrm{T}}$，运用上式就不太方便，须做适当调整。如果规定法向力以外法线方向为正，切向力以沿单元边界前进时，单元保持在左侧为正，如图 6-4 所示。则有关系式

$$p_{sx}=\tau\,\mathrm{d}x/\mathrm{d}s+\sigma\,\mathrm{d}y/\mathrm{d}s,\quad p_{sy}=\tau\,\mathrm{d}y/\mathrm{d}s-\sigma\,\mathrm{d}x/\mathrm{d}s \tag{6-34}$$

把上式代入式(6-29)，就可将第一类曲线积分化为第二类曲线积分式

$$\boldsymbol{f}_{si}^{e}=\begin{Bmatrix}f_{sxi}\\f_{syi}\end{Bmatrix}=\int_{\Gamma}N_i\begin{Bmatrix}\tau\mathrm{d}x+\sigma\mathrm{d}y\\\tau\mathrm{d}y-\sigma\mathrm{d}x\end{Bmatrix}h \quad (i\text{ 取 }\Gamma\text{ 边上的结点}) \tag{6-35}$$

例如 Γ 是图 6-4 中的 374 边，则

$$\boldsymbol{f}_{si}^{e}=\begin{Bmatrix}f_{sxi}\\f_{syi}\end{Bmatrix}=-\int_{-1}^{1}\left(N_i\begin{Bmatrix}\tau x_{,\xi}+\sigma y_{,\xi}\\\tau y_{,\xi}-\sigma x_{,\xi}\end{Bmatrix}\right)_{\eta=1}h\mathrm{d}\xi \quad (i=3,4,7) \tag{6-36}$$

如果法向力 σ 和切向力 τ 也采用形函数插值来近似，那么式(6.36)中的被积函数是多项式，可以得到解析式。当然也可以采用数值积分，因为选择合适的高斯积分阶数，能得到多项式积分的精确值。式(6-36)在实际使用时也不是很方便，因为必须指定某个单元的哪一个边，这样才能确定是 $\xi=1$ 还是 $\eta=1$，是对 ξ 积分还是对 η 积分，还要考虑正负号。而且一般情况下希望用结点号来指定受表面力的边。

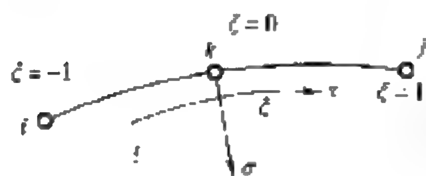


图 6-5 受表面力作用的一条边

实际上，可以把作用表面力的边独立出来，并不需要知道它是单元的哪一条边。设单元的某一条边上作用有表面力 $[\sigma \quad \tau]^{\mathrm{T}}$，如图 6-5 所示，它用 3 个结点($i,k,j$)定义。切向力的正方向为从 i 指向 j，法向力的正方向为切向力顺时针旋转 90°。并沿边的切线方向定义参考坐标 ξ，正方向与切向力相同，可把这条边映射成参考坐标系下长度为 2 的直线。i,k 和 j 点的参考坐标分别为 −1,0 和 1。设 (x_i,y_i)、(x_k,y_k) 和 (x_j,y_j) 是 3 个结点在整体坐标系下的坐标，则这条边的参数方程可写成

$$x = N_i x_i + N_j x_j + N_k x_k, \quad y = N_i y_i + N_j y_j + N_k y_k \tag{6-37}$$

式中

$$N_i = \frac{1}{2}(1-\xi) - \frac{1}{2}N_k, \quad N_j = \frac{1}{2}(1+\xi) - \frac{1}{2}N_k, \quad N_k = 1-\xi^2 \tag{6-38}$$

如果中结点 k 不存在，则令 $N_k=0$，式(6-37)仍然成立。在3个结点上的等效结点力为

$$f_{si}^e = \begin{Bmatrix} f_{sxi} \\ f_{syi} \end{Bmatrix} = \int_{-1}^{1} N_i \begin{Bmatrix} \tau x_{,\xi} + \sigma y_{,\xi} \\ \tau y_{,\xi} - \sigma x_{,\xi} \end{Bmatrix} h \mathrm{d}\xi \quad (i,j,k) \tag{6-39}$$

式中

$$x_{,\xi} = N_{i,\xi} x_i + N_{j,\xi} x_j + N_{k,\xi} x_k, \quad y_{,\xi} = N_{i,\xi} y_i + N_{j,\xi} y_j + N_{k,\xi} y_k \tag{6-40}$$

$$N_{i,\xi} = -\frac{1}{2} - \frac{1}{2}N_{k,\xi}, \quad N_{j,\xi} = \frac{1}{2} - \frac{1}{2}N_{k,\xi}, \quad N_{k,\xi} = -2\xi \tag{6-41}$$

如果法向力 σ 和切向力 τ 也采用形函数插值来近似，即

$$\sigma = N_i\sigma_i + N_k\sigma_k + N_j\sigma_j, \quad \tau = N_i\tau_i + N_k\tau_k + N_j\tau_j \tag{6-42}$$

式中带下标 i、j 和 k 的法向力 σ 和切向力 τ 分别是3个结点处的分布力大小，那么可以得到等效结点力的显式为

$$f_s^e = \begin{Bmatrix} f_{xi} \\ f_{yi} \\ f_{xj} \\ f_{yj} \\ f_{xk} \\ f_{yk} \end{Bmatrix} = \frac{h}{30} \begin{bmatrix} X_1 & X_2 & X_3 & Y_1 & Y_2 & Y_3 \\ Y_1 & Y_2 & Y_3 & -X_1 & -X_2 & -X_3 \\ X_2 & X_4 & X_5 & Y_2 & Y_4 & Y_5 \\ Y_2 & Y_4 & Y_5 & -X_2 & -X_4 & -X_5 \\ X_3 & X_5 & X_6 & Y_3 & Y_5 & Y_6 \\ Y_3 & Y_5 & Y_6 & -X_3 & -X_5 & -X_6 \end{bmatrix} \begin{Bmatrix} \tau_i \\ \tau_j \\ \tau_k \\ \sigma_i \\ \sigma_j \\ \sigma_k \end{Bmatrix} \tag{6-43}$$

式中 X_m 和 Y_m ($m=1,2,\cdots,6$)是与结点坐标有关的常数，即

$$\begin{aligned} &X_1 = -10x_i - 2x_j + 12x_k, \quad X_2 = x_i - x_j, \quad X_3 = -6x_i - 2x_j + 8x_k \\ &Y_1 = -10y_i - 2y_j + 12y_k, \quad Y_2 = y_i - y_j, \quad Y_3 = -6y_i - 2y_j + 8y_k \\ &X_4 = 2x_i + 10x_j - 12x_k, \quad X_5 = 2x_i + 6x_j - 8x_k, \quad X_6 = 16x_i + 16x_j \\ &Y_4 = 2y_i + 10y_j - 12y_k, \quad Y_5 = 2y_i + 6y_j - 8y_k, \quad Y_6 = -16y_i + 16y_j \end{aligned} \tag{6-44}$$

没有中结点 k 时，式(6-43)变成

$$f_s^e = \begin{Bmatrix} f_{xi} \\ f_{yi} \\ f_{xj} \\ f_{yj} \end{Bmatrix} = \frac{h}{6} \begin{bmatrix} -2X_2 & -X_2 & -2Y_2 & -Y_2 \\ -2Y_2 & -Y_2 & 2X_2 & X_2 \\ -X_2 & -2X_2 & -Y_2 & -2Y_2 \\ -Y_2 & -2Y_2 & X_2 & 2X_2 \end{bmatrix} \begin{Bmatrix} \tau_i \\ \tau_j \\ \sigma_i \\ \sigma_j \end{Bmatrix} \tag{6-45}$$

这里利用公式（6-43）给出了计算分布法向力的等效结点力的程序段。注意程序里没有出现单元的厚度，可以理解成这里给的参数 **pressure** 中已经考虑，即它已经根据厚度换算成线分布荷载。

```
1.  function edf = EquivalentDistPressure( node, pressure )
2.  %   计算分布压力的等效结点力
3.  %   输入参数
4.  %     node -------------  结点号
5.  %     pressure -------  与结点号对应的压力值
6.  %   返回值：
7.  %     edf -------------  等效结点力向量
8.      global gNode
9.      x = gNode( node, 1 ) ;
10.     y = gNode( node, 2 ) ;
11.     xi = x(1) ; xj = x(2) ; xk = x(3) ;
12.     yi = y(1) ; yj = y(2) ; yk = y(3) ;
13.     sigi = pressure(1) ;
14.     sigj = pressure(2) ;
15.     sigk = pressure(3) ;
16.     X1 = -10* xi -2* xj +12* xk ;
17.     Y1 = -10* yi -2* yj +12* yk ;
18.     X2 = xi - xj ;
19.     Y2 = yi - yj ;
20.     X3 = -6* xi -2* xj +8* xk ;
21.     Y3 = -6* yi -2* yj +8* yk ;
22.     X4 = 2* xi +10* xj -12* xk ;
23.     Y4 = 2* yi +10* yj -12* yk ;
24.     X5 = 2* xi +6* xj -8* xk ;
25.     Y5 = 2* yi +6* yj -8* yk ;
26.     X6 = -16* xi +16* xj ;
27.     Y6 = -16* yi +16* yj ;
28.     XY = [  Y1    Y2    Y3
29.            -X1   -X2   -X3
30.             Y2    Y4    Y5
31.            -X2   -X4   -X5
32.             Y3    Y5    Y6
```

```
33.            -X3   -X5   -X6];
34. [illegible]
35. [illegible]
36.   return
```

4. 温度荷载

如果考虑温度改变引起的初应变 $\boldsymbol{\varepsilon}_0 = \alpha T[1 \quad 1 \quad 0]^{\mathrm{T}}$，与以前相类似，需要增加由于温度改变所产生的等效结点力，即

$$f_T^e = \iint_{\Omega} \boldsymbol{B}^{\mathrm{T}} \boldsymbol{D} \boldsymbol{\varepsilon}_0 \, h \mathrm{d}x \mathrm{d}y \tag{6-46}$$

对于平面应力问题得到

$$f_{Ti} = \frac{E\alpha}{1-\mu} \iint_A \begin{Bmatrix} N_{i,x} \\ N_{i,y} \end{Bmatrix} hT \mathrm{d}x\mathrm{d}y = \frac{E\alpha h}{1-\mu} \int_{-1}^{1} \int_{-1}^{1} \begin{Bmatrix} N_{i,x} \\ N_{i,y} \end{Bmatrix} T |\boldsymbol{J}| \mathrm{d}\xi \mathrm{d}\eta \quad (i = 1,2,\cdots,n) \tag{6-47}$$

此时，计算应力的公式(6-9)要改写为

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{B}\boldsymbol{\delta}^e - \boldsymbol{D}\boldsymbol{\varepsilon}_0 \tag{6-48}$$

6.1.6 应力的计算

四边形等参元中，单元应变和应力不再是常数，因为应变矩阵 $\boldsymbol{B}$ 和应力矩阵 $\boldsymbol{S}$ 都是参考坐标 ξ 和 η 的函数。一般情况下，高斯点处的应力精度较高。对于 2×2 积分的四边形，可得到 4 组应力，但是，结点上的应力更加实用，这是因为结点常常位于物体的表面，它们上面的应力值一般要大于结构内部。另外，一个结点经常被几个单元共享，由各个单元分别计算得到的结点应力差异，可以用来度量分析的误差。单元内结点应力计算可直接把结点处的参考坐标值代入应力矩阵利用公式(6-9)计算，也可以先计算高斯点处的应力，然后外插得到[5]。一般情况下，单元间的应力是不连续的，因此可以采用结点平均、单元平均、总体磨平或局部磨平等方法进行处理[7]，以提高应力的计算精度。

6.1.7 单元的完备性和协调性

由于相邻单元的交界线只与交界线上的结点坐标有关，而且交界线上的位移只与该线上的结点位移有关，而与其他结点位移无关，因此，交界线上的位移由该线上的结点位移所惟一确定，故单元的协调性得到满足。

根据前面的讨论知道，在平面问题中，对应于刚体位移和常应变状态的位移可写成如下形式

$$u = a_0 + a_1 x + a_2 y, \quad v = b_0 + b_1 x + b_2 y \tag{6-49}$$

与这个单元位移相对应的结点位移，显然只要把相应的结点坐标代入上式即可。现假设等参数单元的结点位移是由上式所示的刚体位移或常应变所对应的位移所确定，即

$$u_i = a_0 + a_1 x_i + a_2 y_i, \quad v_i = b_0 + b_1 x_i + b_2 y_i \tag{6-50}$$

单元内的位移根据式(6-1)，由结点位移插值得到。当结点位移如上式所示时，单元内位移为

$$u = \sum_{i=1}^{n} N_i (a_0 + a_1 x_i + a_2 y_i) = a_0 \sum_{i=1}^{n} N_i + a_1 \sum_{i=1}^{n} N_i x_i + a_2 \sum_{i=1}^{n} N_i y_i \tag{6-51}$$

$$v = \sum_{i=1}^{n} N_i (b_0 + b_1 x_i + b_2 y_i) = b_0 \sum_{i=1}^{n} N_i + b_1 \sum_{i=1}^{n} N_i x_i + b_2 \sum_{i=1}^{n} N_i y_i \tag{6-52}$$

由于形函数之和为 1 以及注意到式（6-2)，则单元内的位移就成为式(6-49)所示的形式，即单元内的位移是刚体位移或者对应于常应变的位移，可见完备性得到满足。因此等参元是完备的协调单元。

最后指出，前面两章中所讨论的三角形单元和四面体单元，其实也都是等参数单元。例如，三角形单元的坐标变换式是

$$x = L_i x_i + L_j x_j + L_m x_m \tag{6-53}$$

$$y = L_i y_i + L_j x_j + L_m x_m \tag{6-54}$$

而位移模式是

$$u = L_i u_i + L_j u_j + L_m u_m \tag{6-55}$$

$$v = L_i v_i + L_j v_j + L_m v_m \tag{6-56}$$

它们的坐标变换和形函数是相同的，所以也是等参数单元。

6.2 空间轴对称等参数单元

6.2.1 单元特性

在空间轴对称问题中，采用的整体坐标系是圆柱坐标系，r表示径向坐标

z 表示轴向坐标，而 u 和 w 分别表示径向和轴向位移。和前面相同，坐标变换式和位移模式分别具有下列形式

$$r = \sum_{i=1}^{n} N_i r_i, \quad z = \sum_{i=1}^{n} N_i z_i \tag{6-57}$$

$$u = \sum_{i=1}^{n} N_i u_i, \quad w = \sum_{i=1}^{n} N_i w_i \tag{6-58}$$

式中，$n=4\sim8$ 是单元的结点数，形函数 N_i 见式(6-15)～式(6-22)。

应变计算公式是

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \partial u/\partial r \\ u/r \\ \partial w/\partial r \\ \partial u/\partial z + \partial w/\partial r \end{Bmatrix} = \boldsymbol{B}\boldsymbol{\delta}^e = [\boldsymbol{B}_1 \quad \boldsymbol{B}_2 \quad \cdots \quad \boldsymbol{B}_n] \begin{Bmatrix} \boldsymbol{\delta}_1 \\ \boldsymbol{\delta}_2 \\ \vdots \\ \boldsymbol{\delta}_n \end{Bmatrix} \tag{6-59}$$

其中

$$\boldsymbol{B}_i = \begin{bmatrix} N_{i,r} & 0 \\ N_{i/r} & 0 \\ 0 & N_{i,z} \\ N_{i,z} & N_{i,r} \end{bmatrix}, \quad \boldsymbol{\delta}_i = \begin{Bmatrix} u_i \\ w_i \end{Bmatrix} \; (i=1,2,\cdots,n) \tag{6-60}$$

其中，$N_{i,r}$ 和 $N_{i,z}$ 分别表示 N_i 对 r 和 z 的偏导数。它们可按下式计算

$$\begin{Bmatrix} N_{i,r} \\ N_{i,z} \end{Bmatrix} = \boldsymbol{J}^{-1} \begin{Bmatrix} N_{i,\xi} \\ N_{i,\eta} \end{Bmatrix} \tag{6-61}$$

而 $N_{i,\xi}$ 和 $N_{i,\eta}$ 分别由形函数对 ξ 和 η 求导数得出，矩阵 $\boldsymbol{J}^{-1}$ 是雅可比矩阵 $\boldsymbol{J}$ 的逆阵

$$\boldsymbol{J}^{-1} = \frac{1}{|\boldsymbol{J}|} \begin{bmatrix} z_{,\eta} & -z_{,\xi} \\ -r_{,\eta} & r_{,\xi} \end{bmatrix} \tag{6-62}$$

雅可比行列式是

$$|\boldsymbol{J}| = r_{,\xi} z_{,\eta} - r_{,\eta} z_{,\xi} \tag{6-63}$$

式中

$$r_{,\xi} = \sum_{i=1}^{n} N_{i,\xi} r_i, \quad r_{,\eta} = \sum_{i=1}^{n} N_{i,\eta} r_i \tag{6-64}$$

$$z_{,\xi} = \sum_{i=1}^{n} N_{i,\xi} z_i, \quad z_{,\eta} = \sum_{i=1}^{n} N_{i,\eta} z_i \tag{6-65}$$

应力计算公式是

$$\boldsymbol{\sigma}=\begin{Bmatrix}\sigma_r\\ \sigma_\theta\\ \sigma_z\\ \tau_{rz}\end{Bmatrix}=\boldsymbol{DB\delta}^e=[\boldsymbol{DB}_1\quad \boldsymbol{DB}_2\quad \cdots\quad \boldsymbol{DB}_n]\boldsymbol{\delta}^e \tag{6-66}$$

对于各向同性材料，式中

$$\boldsymbol{DB}_i=A_3\begin{bmatrix} N_{i,r}+A_1 N_i/r & A_1 N_{i,z}\\ A_1 N_{i,r}+N_i/r & A_1 N_{i,z}\\ A_1(N_{i,r}+N_i/r) & N_{i,z}\\ A_2 N_{i,z} & A_2 N_{i,r}\end{bmatrix}\quad (i=1,2,\cdots,n) \tag{6-67}$$

式中 A_1、A_2 和 A_3 由式(5-17)定义。当 $r=0$ 时，即在对称轴上有 $\varepsilon_r=\varepsilon_\theta$，因此可以用 $N_{i,r}$ 代替 N_i/r 以消除式(6-67)中的奇异项。

单元刚度矩阵可以分成 $n\times n$ 个子矩阵，典型的子矩阵是

$$\boldsymbol{K}_{ij}=2\pi\int_{-1}^{1}\int_{-1}^{1}\boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{DB}_j\, r\,|\boldsymbol{J}|\,\mathrm{d}\xi\mathrm{d}\eta \tag{6-68}$$

式中

$$\boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{DB}_j=A_3\left[\begin{array}{c:c} N_{i,r}(N_{j,r}+A_1 N_j/r) & A_1 N_{i,r}N_{j,z}\\ +N_i(A_1 N_{j,r}+N_j/r)/r & +A_1 N_i N_{j,z}/r\\ +A_2 N_{i,z}N_{j,z} & +A_2 N_{i,z}N_{j,r}\\ \hdashline A_1 N_{i,z}(N_{j,r}+N_j/r) & N_{i,z}N_{j,z}\\ +A_2 N_{i,r}N_{j,z} & +A_2 N_{i,r}N_{j,r}\end{array}\right]$$

$$(i,j=1,2,\cdots,n) \tag{6-69}$$

6.2.2 等效结点力

1. 体积力

设单位体积的体积力是 $\boldsymbol{p}_v=[p_{vr}\quad p_{vz}]^{\mathrm{T}}$，则作用在单元各结点上的等效结点力分别按下式计算

$$\boldsymbol{f}_{vi}=\begin{Bmatrix}f_{vri}\\ f_{vzi}\end{Bmatrix}=2\pi\int_{-1}^{1}\int_{-1}^{1} rN_i\begin{Bmatrix}p_{vr}\\ p_{vz}\end{Bmatrix}|\boldsymbol{J}|\,\mathrm{d}\xi\mathrm{d}\eta\quad (i=1,2,\cdots,n) \tag{6-70}$$

2. 表面力

设沿单元的某边界 $\boldsymbol{\Gamma}$ 上作用有表面力 $\boldsymbol{p}_s=[\sigma\quad \tau]^{\mathrm{T}}$，其中 σ 和 τ 分别是单元表面力在作用边外法线方向和切线方向的投影。于是，作用在此边上各

结点上的等效结点力分别按下式计算

$$f_{si} = \begin{Bmatrix} f_{sri} \\ f_{szi} \end{Bmatrix} = 2\pi \int_{\Gamma} r N_i \begin{Bmatrix} \tau \mathrm{d}r + \sigma \mathrm{d}z \\ \tau \mathrm{d}z - \sigma \mathrm{d}r \end{Bmatrix} \quad i \text{ 取 } \Gamma \text{ 上的结点} \tag{6-71}$$

例如，Γ 是对应 $\eta = 1$ 的那条边，上式可以写成如下形式

$$f_{si} = \begin{Bmatrix} f_{sri} \\ f_{szi} \end{Bmatrix} = -2\pi \int_{-1}^{1} \left(r N_i \begin{Bmatrix} \tau r_{,\xi} + \sigma z_{,\xi} \\ \tau z_{,\xi} - \sigma r_{,\xi} \end{Bmatrix} \right)_{\eta = 1} \mathrm{d}\xi \quad i \text{ 取 } \Gamma \text{ 上的结点} \tag{6-72}$$

为了方便，跟前面平面等参数单元一样，可以只用 3 个结点(i,j,k) 来表示受力的那条边，切向力和法向力的正方向也跟平面等参元规定的一样，那么这 3 个上的等效结点力公式可写为

$$f_{si} = \begin{Bmatrix} f_{sri} \\ f_{szi} \end{Bmatrix} = 2\pi \int_{-1}^{1} r N_i \begin{Bmatrix} \tau r_{,\xi} + \sigma z_{,\xi} \\ \tau z_{,\xi} - \sigma r_{,\xi} \end{Bmatrix} \mathrm{d}\xi \quad (i,j,k) \tag{6-73}$$

式中整体坐标对局部坐标的导数为

$$r_{,\xi} = N_{i,\xi} r_i + N_{j,\xi} r_j + N_{k,\xi} r_k, \quad z_{,\xi} = N_{i,\xi} z_i + N_{j,\xi} z_j + N_{k,\xi} z_k \tag{6-74}$$

而上面两式中用到的形函数由式（6-38）定义。

3. 温度荷载

如果考虑温度变化产生的初应变

$$\varepsilon_0 = \alpha T [1 \quad 1 \quad 1 \quad 0]^{\mathrm{T}} \tag{6-75}$$

则单元结点上等效结点力的计算公式是

$$f_{Ti} = \begin{Bmatrix} f_{Tri} \\ f_{Tzi} \end{Bmatrix} = 2\pi \iint B_i^{\mathrm{T}} D \varepsilon_0 \, r \mathrm{d}r \mathrm{d}z = \frac{2E\alpha\pi}{1 - 2\mu} \int_{-1}^{1} \int_{-1}^{1} r T \begin{Bmatrix} N_{i,r} + N_i / r \\ N_{i,z} \end{Bmatrix} |J| \mathrm{d}\xi \mathrm{d}\eta \tag{6-76}$$

计算应力的公式(6-66)要改写为

$$\sigma = DB\delta^e - D\varepsilon_0 = [DB_1 \quad DB_2 \quad \cdots \quad DB_n]\delta^e - \frac{E\alpha T}{1 - 2\mu}[1 \quad 1 \quad 1 \quad 0]^{\mathrm{T}} \tag{6-77}$$

这些公式与平面问题的四边形单元相比，只是被积函数中多了变量 r，积分号外多了 2π，对于数值积分就没有任何困难，可仿照平面单元进行。

6.3 空间等参数单元

6.3.1 单元特性

空间等参数单元，可由平面问题等参数单元推广得到。平面的 4 结点单元对应于空间的 8 结点单元，平面的 8 结点单元对应于空间的 20 结点单元。空间 8 结点单元是直棱的六面体，而 20 结点的则可以是曲面曲棱的六面体，可以描述形状复杂的三维结构。一个 20 结点等参数单元及其参考坐标系 $\xi\eta\zeta$ 如图 6-6 所示。在参考坐标系下，曲面曲棱的六面体被变换成一个边长为 2 的立方体，参考坐标系 $\xi\eta\zeta$ 的原点位于它的形心处。

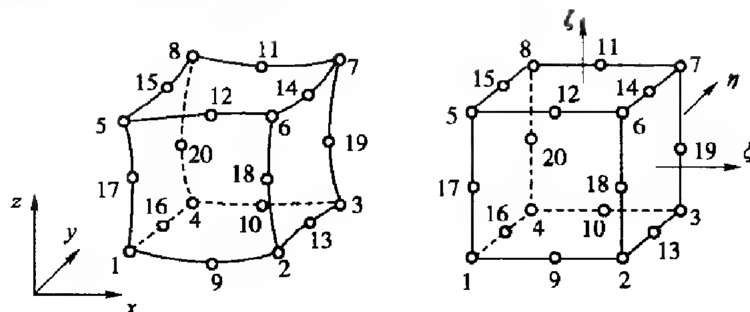


图 6-6 20 结点等参数单元

坐标变换式和位移模式可统一写成如下形式

$$x = \sum_{i=1}^{n} N_i x_i, \quad y = \sum_{i=1}^{n} N_i y_i, \quad z = \sum_{i=1}^{n} N_i z_i \tag{6-78}$$

$$u = \sum_{i=1}^{n} N_i u_i, \quad v = \sum_{i=1}^{n} N_i v_i, \quad w = \sum_{i=1}^{n} N_i w_i \tag{6-79}$$

式中 n 是结点数。当 $n=8$ 时，指的是 8 结点等参数单元，首先来写出它的形函数

$$N_i = (1+\xi_i\xi)(1+\eta_i\eta)(1+\zeta_i\zeta)/8 \quad (i=1,2,\cdots,8) \tag{6-80}$$

式中 ξ_i、η_i 和 ζ_i 是结点 i 的局部坐标，对于角结点它们分别为 1 或 −1，即

$$\begin{aligned}
\xi_{1,2,\cdots,8} &= -1, \quad 1, \quad 1, \quad -1, \quad -1, \quad 1, \quad 1, \quad -1 \\
\eta_{1,2,\cdots,8} &= -1, \quad -1, \quad 1, \quad 1, \quad -1, \quad -1, \quad 1, \quad 1 \\
\zeta_{1,2,\cdots,8} &= -1, \quad -1, \quad -1, \quad -1, \quad 1, \quad 1, \quad 1, \quad 1
\end{aligned} \tag{6-81}$$

观察形函数式(6-80)，其右端的每一项正好是距结点 i 的距离为 2 的三

个平面方程，因此将其他 7 个角结点坐标代入有等于零的结果。将结点 i 的坐标(ξ_i,η_i,ζ_i)代入则正好等于 1，因此系数八分之一是按形函数的要求而确定的。仿照这个办法，能写出结点 9 到 20 对应的形函数

$$
\begin{aligned}
N_i &= (1-\xi^2)(1+\eta_i\eta)(1+\zeta_i\zeta)/4 \quad (i=9,10,11,12)\\
N_i &= (1-\eta^2)(1+\xi_i\xi)(1+\zeta_i\zeta)/4 \quad (i=13,14,15,16)\\
N_i &= (1-\zeta^2)(1+\xi_i\xi)(1+\eta_i\eta)/4 \quad (i=17,18,19,20)
\end{aligned} \tag{6-82}
$$

对于结点 9 到 20,ξ_i,η_i 和 ζ_i 分别取 1、-1 或者 0。即

$$
\begin{aligned}
\xi_{9,10,\cdots,20} &= \ \ 0,\ \ 0,\ \ 0,\ \ 0,\ \ 1,\ \ 1,\ -1,\ -1,\ -1,\ \ 1,\ \ 1,\ -1\\
\eta_{9,10,\cdots,20} &= -1,\ \ 1,\ \ 1,\ -1,\ \ 0,\ \ 0,\ \ 0,\ \ 0,\ -1,\ -1,\ \ 1,\ \ 1\\
\zeta_{9,10,\cdots,20} &= -1,\ -1,\ \ 1,\ \ 1,\ -1,\ \ 1,\ \ 1,\ -1,\ \ 0,\ \ 0,\ \ 0,\ \ 0
\end{aligned} \tag{6-83}
$$

对于 8 结点等参数单元，其形函数为式(6-80)。对于 20 结点等参数单元，其边中结点形函数如式(6-82)所示，而角结点的形函数由如下的线性组合得出

$$
N_i = \overline{N}_i - \sum_{j=1}^{12} \overline{N}_i(\xi_{j+8},\eta_{j+8},\zeta_{j+8})N_{j+8} \quad (i=1,2,\cdots,8) \tag{6-84}
$$

式中 $\overline{N}_i$ 即为式(6-80)定义的形函数。实际上，上式具体写出来更容易记忆，即

$$
\begin{aligned}
N_1 &= \overline{N}_1 - \frac{1}{2}(N_9+N_{16}+N_{17}), \quad N_2 = \overline{N}_2 - \frac{1}{2}(N_9+N_{13}+N_{18})\\
N_3 &= \overline{N}_3 - \frac{1}{2}(N_{10}+N_{13}+N_{19}), \quad N_4 = \overline{N}_4 - \frac{1}{2}(N_{10}+N_{16}+N_{20})\\
N_5 &= \overline{N}_5 - \frac{1}{2}(N_{12}+N_{15}+N_{17}), \quad N_6 = \overline{N}_6 - \frac{1}{2}(N_{12}+N_{14}+N_{18})\\
N_7 &= \overline{N}_7 - \frac{1}{2}(N_{11}+N_{14}+N_{19}), \quad N_8 = \overline{N}_8 - \frac{1}{2}(N_{11}+N_{15}+N_{20})
\end{aligned} \tag{6-85}
$$

即 20 结点单元角结点的形函数是 8 结点的形函数减去与该结点最近的 3 个边中结点形函数之和的一半。如果增加一个约定，在形函数(6-82)中令某一个形函数或某几个形函数恒等于零，即表示 20 结点单元有相应一个或几个边中结点不存在。有了这个约定，则式(6-80)和式(6-85)就可作为 8～20 结点

等参数单元的形函数。

按几何关系式和式(6-79),应变计算公式是

$$\boldsymbol{\varepsilon}=\begin{Bmatrix}\partial u/\partial x\\ \partial v/\partial y\\ \partial w/\partial z\\ \partial w/\partial y+\partial v/\partial z\\ \partial w/\partial x+\partial u/\partial z\\ \partial v/\partial x+\partial u/\partial y\end{Bmatrix}=\boldsymbol{B}\boldsymbol{\delta}^{e}=[\boldsymbol{B}_1\quad \boldsymbol{B}_2\quad \cdots\quad \boldsymbol{B}_n]\begin{Bmatrix}\boldsymbol{\delta}_1\\ \boldsymbol{\delta}_2\\ \vdots\\ \boldsymbol{\delta}_n\end{Bmatrix}\tag{6-86}$$

其中

$$\boldsymbol{B}_i=\begin{bmatrix}N_{i,x} & 0 & 0\\ 0 & N_{i,y} & 0\\ 0 & 0 & N_{i,z}\\ 0 & N_{i,z} & N_{i,y}\\ N_{i,z} & 0 & N_{i,x}\\ N_{i,y} & N_{i,x} & 0\end{bmatrix},\quad \boldsymbol{\delta}_i=\begin{Bmatrix}u_i\\ v_i\\ w_i\end{Bmatrix}\quad (i=1,2,\cdots,n)\tag{6-87}$$

记号 $\boldsymbol{N}_{i,x}$、$\boldsymbol{N}_{i,y}$ 和 $\boldsymbol{N}_{i,z}$ 分别表示 $\boldsymbol{N}_i$ 对 x、y 和 z 的偏导数。根据复合函数求导的规则,它们与 $\boldsymbol{N}_{i,\xi}$、$\boldsymbol{N}_{i,\eta}$ 和 $\boldsymbol{N}_{i,\zeta}$ 有如下关系式

$$\begin{Bmatrix}N_{i,\xi}\\ N_{i,\eta}\\ N_{i,\zeta}\end{Bmatrix}=\begin{bmatrix}x_{,\xi} & y_{,\xi} & z_{,\xi}\\ x_{,\eta} & y_{,\eta} & z_{,\eta}\\ x_{,\zeta} & y_{,\zeta} & z_{,\zeta}\end{bmatrix}\begin{Bmatrix}N_{i,x}\\ N_{i,y}\\ N_{i,z}\end{Bmatrix}=\boldsymbol{J}\begin{Bmatrix}N_{i,x}\\ N_{i,y}\\ N_{i,z}\end{Bmatrix}\tag{6-88}$$

式中

$$x_{,\xi}=\sum_{i=1}^{n}N_{i,\xi}x_i,\cdots,z_{,\zeta}=\sum_{i=1}^{n}N_{i,\zeta}z_i\tag{6-89}$$

于是按式(6-88)求逆,得到

$$\begin{Bmatrix}N_{i,x}\\ N_{i,y}\\ N_{i,z}\end{Bmatrix}=\boldsymbol{J}^{-1}\begin{Bmatrix}N_{i,\xi}\\ N_{i,\eta}\\ N_{i,\zeta}\end{Bmatrix}\tag{6-90}$$

为了用 $\boldsymbol{J}$ 的元素写出 $\boldsymbol{J}^{-1}$ 的显式,引入矢径及其偏导数的记号如下

$$\boldsymbol{r}=[x\quad y\quad z]^{\mathrm{T}},\quad \boldsymbol{s}=\boldsymbol{r}_{,\xi}=[x_{,\xi}\quad y_{,\xi}\quad z_{,\xi}]^{\mathrm{T}}\tag{6-91}$$

$$\boldsymbol{t}=\boldsymbol{r}_{,\eta}=[x_{,\eta}\quad y_{,\eta}\quad z_{,\eta}]^{\mathrm{T}},\quad \boldsymbol{v}=\boldsymbol{r}_{,\zeta}=[x_{,\zeta}\quad y_{,\zeta}\quad z_{,\zeta}]^{\mathrm{T}} \tag{6-92}$$

则

$$\boldsymbol{J}=[\boldsymbol{s}\quad \boldsymbol{t}\quad \boldsymbol{v}]^{\mathrm{T}} \tag{6-93}$$

$$|\boldsymbol{J}|=\boldsymbol{s}\cdot\boldsymbol{t}\times\boldsymbol{v}=\boldsymbol{t}\cdot\boldsymbol{v}\times\boldsymbol{s}=\boldsymbol{s}\times\boldsymbol{t}\cdot\boldsymbol{v} \tag{6-94}$$

$$\boldsymbol{J}^{-1}=[\boldsymbol{t}\times\boldsymbol{v}\quad \boldsymbol{v}\times\boldsymbol{s}\quad \boldsymbol{s}\times\boldsymbol{t}]/|\boldsymbol{J}| \tag{6-95}$$

应力的计算公式是

$$\begin{aligned}\boldsymbol{\sigma}&=[\sigma_x\quad \sigma_y\quad \sigma_z\quad \tau_{yz}\quad \tau_{xz}\quad \tau_{xy}]^{\mathrm{T}}\\&=\boldsymbol{DB\delta}^e=[\boldsymbol{DB}_1\quad \boldsymbol{DB}_2\quad \cdots\quad \boldsymbol{DB}_n]\boldsymbol{\delta}^e\end{aligned} \tag{6-96}$$

而

$$\boldsymbol{DB}_i=A_3\begin{bmatrix}N_{i,x} & A_1N_{i,y} & A_1N_{i,z}\\ A_1N_{i,x} & N_{i,y} & A_1N_{i,z}\\ A_1N_{i,x} & A_1N_{i,y} & N_{i,z}\\ 0 & A_2N_{i,z} & A_2N_{i,y}\\ A_2N_{i,z} & 0 & A_2N_{i,x}\\ A_2N_{i,y} & A_2N_{i,x} & 0\end{bmatrix}\quad(i=1,2,\cdots,n) \tag{6-97}$$

其中常数 A_1、A_2和 A_3 由式(5-17)定义。

单元刚度矩阵可以分成 $n\times n$ 个子矩阵，典型的子矩阵是

$$\boldsymbol{K}_{ij}=\iiint_V \boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{DB}_j\,\mathrm{d}x\mathrm{d}y\mathrm{d}z=\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1}\boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{DB}_j\,|\boldsymbol{J}|\,\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta$$
$$(i,j=1,2,\cdots,n) \tag{6-98}$$

而

$$\boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{DB}_j=A_3\left[\begin{array}{c:c:c}\begin{gathered}N_{i,x}N_{j,x}\\+A_2(N_{i,z}N_{j,z}\\+N_{i,y}N_{j,y})\end{gathered} & \begin{gathered}A_1N_{i,x}N_{j,y}\\+A_2N_{i,y}N_{j,x}\end{gathered} & \begin{gathered}A_1N_{i,x}N_{j,z}\\+A_2N_{i,z}N_{j,x}\end{gathered}\\ \hdashline \begin{gathered}A_1N_{i,y}N_{j,x}\\+A_2N_{i,x}N_{j,y}\end{gathered} & \begin{gathered}N_{i,y}N_{j,y}\\+A_2(N_{i,z}N_{j,z}\\+N_{i,x}N_{j,x})\end{gathered} & \begin{gathered}A_1N_{i,y}N_{j,z}\\+A_2N_{i,z}N_{j,y}\end{gathered}\\ \hdashline \begin{gathered}A_1N_{i,z}N_{j,x}\\+A_2N_{i,x}N_{j,z}\end{gathered} & \begin{gathered}A_1N_{i,z}N_{j,y}\\+A_2N_{i,y}N_{j,z}\end{gathered} & \begin{gathered}N_{i,z}N_{j,z}\\+A_2(N_{i,y}N_{j,y}\\+N_{i,x}N_{j,x})\end{gathered}\end{array}\right]$$
$$(i,j=1,2,\cdots,n) \tag{6-99}$$

下面是计算单元刚度矩阵的函数，其中又调用了计算应变矩阵，Jacobi 矩阵，形函数对坐标的导数等子程序。为了方便阅读，这里一并列出。

```
1.  function k = StiffnessMatrix( ie )
2.  %  计算三维 20 结点等参数单元的刚度矩阵
3.  %  输入参数:
4.  %      ie ---- 单元号
5.  %  返回值:
6.  %      k ---- 单元刚度矩阵
7.  %  说明:
8.  %      用高斯积分法求解平面等参数单元的刚度矩阵
9.
10.     k = zeros( 60, 60 ) ;
11.     D = MatrixD( ie ) ;
12.     % 3 x 3  高斯积分点和权系数
13.     x = [-0.774596669241483,           0.0,0.774596669241483] ;
14.     w = [ 0.555555555555556,0.888888888888889,0.555555555555556] ;
15.     for i=1:1:length(x)
16.         for j=1:1:length(x)
17.             for m=1:1:length(x)
18.                 B = MatrixB( ie, x(i), x(j), x(m) ) ;
19.                 J = Jacobi( ie, x(i), x(j), x(m) ) ;
20.                 k = k + w(i)*w(j)*w(m)*transpose(B)*D*B*det(J) ;
21.             end
22.         end
23.     end
24. return
25.
26. function D = MatrixD( ie )
27. %  计算单元的弹性矩阵 D
28. %  输入参数:
29. %      ie ------------ 单元号
30. %  返回值:
31. %      D ------------- 弹性矩阵 D
32.
33.     global gElement gMaterial
```

```
34.     E  = gMaterial( gElement(ie, 21), 1 ) ;    % 弹性模量
35.     mu = gMaterial( gElement(ie, 21), 2 ) ;    % 泊松比
36.     A1 = mu/(1-mu) ;
37.     A2 = (1-2*mu)/2/(1-mu) ;
38.     A3 = E*(1-mu)/(1+mu)/(1-2*mu) ;
39.     D = A3*[ 1   A1  A1  0   0   0
40.              A1  1   A1  0   0   0
41.              A1  A1  1   0   0   0
42.              0   0   0   A2  0   0
43.              0   0   0   0   A2  0
44.              0   0   0   0   0   A2] ;
45. return
46.
47. function B = MatrixB( ie, xi, eta, zeta )
48. %  计算单元的应变矩阵B
49. %  输入参数:
50. %     ie --------- 单元号
51. %     xi,eta ----- 局部坐标
52. %  返回值:
53. %     B  --------- 在局部坐标处的应变矩阵B
54.
55. [N_x, N_y, N_z] = N_xyz( ie, xi, eta, zeta );
56.     B = zeros( 6, 60 ) ;
57.     for i=1:20
58.         B(:,(3*i-2):3*i) = [ N_x(i)   0       0
59.                              0     N_y(i)     0
60.                              0       0     N_z(i)
61.                              0     N_z(i)  N_y(i)
62.                            N_z(i)    0     N_x(i)
63.                            N_y(i)  N_x(i)    0];
64.     end
65. return
66.
67. function [N_x, N_y, N_z] = N_xyz( ie, xi, eta, zeta )
68. %  计算形函数对整体坐标的导数
69. %  输入参数:
```

```
70.  %        ie ---------------- 单元号
71.  %        xi,eta,zeta ------- 局部坐标
72.  %   返回值:
73.  %        N_x ------------- 在局部坐标处的形函数对 x 坐标的导数
74.  %        N_y ------------- 在局部坐标处的形函数对 y 坐标的导数
75.  %        N_z ------------- 在局部坐标处的形函数对 z 坐标的导数
76.
77.      J = Jacobi( ie, xi, eta, zeta ) ;
78.      [N_xi,N_eta,N_zeta] = N_xietazeta( ie, xi, eta, zeta ) ;
79.      N = [N_xi;N_eta;N_zeta] ;
80.      A = inv(J)*N ;
81.      N_x = A(1,:) ;
82.      N_y = A(2,:) ;
83.      N_z = A(3,:) ;
84.  return
85.
86.  function [N_xi, N_eta, N_zeta] = ...
         N_xietazeta( ie, xi, eta, zeta)
87.  %   计算形函数对局部坐标的导数
88.  %   输入参数:
89.  %        ie ---------------- 单元号
90.  %        xi,eta,zeta ------- 局部坐标
91.  %   返回值:
92.  %        N_xi ------------- 在局部坐标处的形函数对 xi 坐标的导数
93.  %        N_eta ------------ 在局部坐标处的形函数对 eta 坐标的导数
94.  %        N_zeta ----------- 在局部坐标处的形函数对 zeta 坐标的导数
95.
96.      x = [-1,1,1,-1,-1,1,1,-1,0,0,0,0,-1,1,1,-1,-1,1,1,-1] ;
97.      e = [-1,-1,1,1,-1,-1,1,1,-1,1,1,-1,0,0,0,0,-1,-1,1,1] ;
98.      z = [-1,-1,-1,-1,1,1,1,1,-1,-1,1,1,-1,1,1,-1,0,0,0,0] ;
99.      N_xi   = zeros( 1, 20 ) ;
100.     N_eta  = zeros( 1, 20 ) ;
101.     N_zeta = zeros( 1, 20 ) ;
102.
103.     N_xi(1:8)    = x(1:8).*(1+e(1:8)*eta).*(1+z(1:8)*zeta)/8 ;
104.     N_eta(1:8)   = (1+x(1:8)*xi).*e(1:8).*(1+z(1:8)*zeta)/8 ;
```

```
105.   N_zeta(1:8) = (1+x(1:8)*xi).*(1+e(1:8)*eta).*z(1:8)/8;
106.
107.   N_xi(9:12)   = -2*xi*(1+e(9:12)*eta).*(1+z(9:12)*zeta)/4;
108.   N_eta(9:12)  = (1-xi^2)*e(9:12).*(1+z(9:12)*zeta)/4;
109.   N_zeta(9:12) = (1-xi^2)*(1+e(9:12)*eta).*z(9:12)/4;
110.
111.   N_xi(13:16)  = (1-eta^2)*x(13:16).*(1+z(13:16)*zeta)/4;
112.   N_eta(13:16) = -2*eta*(1+x(13:16)*xi).*(1+z(13:16)*zeta)/4;
113.   N_zeta(13:16) = (1-eta^2)*(1+x(13:16)*xi).*z(13:16)/4;
114.
115.   N_xi(17:20)  = (1-zeta^2)*x(17:20).*(1+e(17:20)*eta)/4;
116.   N_eta(17:20) = (1-zeta^2)*(1+x(17:20)*xi).*e(17:20)/4;
117.   N_zeta(17:20) = -2*zeta*(1+x(17:20)*xi).*(1+e(17:20)*eta)/4;
118.
119.   N_xi(1) = N_xi(1) - 0.5*(N_xi(9)+N_xi(16)+N_xi(17));
120.   N_xi(2) = N_xi(2) - 0.5*(N_xi(9)+N_xi(13)+N_xi(18));
121.   N_xi(3) = N_xi(3) - 0.5*(N_xi(10)+N_xi(13)+N_xi(19));
122.   N_xi(4) = N_xi(4) - 0.5*(N_xi(10)+N_xi(16)+N_xi(20));
123.   N_xi(5) = N_xi(5) - 0.5*(N_xi(12)+N_xi(15)+N_xi(17));
124.   N_xi(6) = N_xi(6) - 0.5*(N_xi(12)+N_xi(14)+N_xi(18));
125.   N_xi(7) = N_xi(7) - 0.5*(N_xi(11)+N_xi(14)+N_xi(19));
126.   N_xi(8) = N_xi(8) - 0.5*(N_xi(11)+N_xi(15)+N_xi(20));
127.
128.   N_eta(1) = N_eta(1) - 0.5*(N_eta(9)+N_eta(16)+N_eta(17));
129.   N_eta(2) = N_eta(2) - 0.5*(N_eta(9)+N_eta(13)+N_eta(18));
130.   N_eta(3) = N_eta(3) - 0.5*(N_eta(10)+N_eta(13)+N_eta(19));
131.   N_eta(4) = N_eta(4) - 0.5*(N_eta(10)+N_eta(16)+N_eta(20));
132.   N_eta(5) = N_eta(5) - 0.5*(N_eta(12)+N_eta(15)+N_eta(17));
133.   N_eta(6) = N_eta(6) - 0.5*(N_eta(12)+N_eta(14)+N_eta(18));
134.   N_eta(7) = N_eta(7) - 0.5*(N_eta(11)+N_eta(14)+N_eta(19));
135.   N_eta(8) = N_eta(8) - 0.5*(N_eta(11)+N_eta(15)+N_eta(20));
136.
137.   N_zeta(1) = N_zeta(1) - 0.5*(N_zeta(9)+N_zeta(16)+...
       N_zeta(17));
138.   N_zeta(2) = N_zeta(2) - 0.5*(N_zeta(9)+N_zeta(13)+...
       N_zeta(18));
```

```
139.    N_zeta(3) = N_zeta(3) - 0.5* (N_zeta(10) +N_zeta(13) +...
        N_zeta(19)) ;
140.    N_zeta(4) = N_zeta(4) - 0.5* (N_zeta(10) +N_zeta([illegible]) +...
        N_zeta(20)) ;
141.    N_zeta(5) = N_zeta(5) - 0.5* (N_zeta([illegible]) +N_zeta([illegible]) +...
        N_zeta(17)) ;
142.    N_zeta(6) = N_zeta(6) - 0.5* (N_zeta([illegible]) +N_zeta([illegible]) +...
        N_zeta(18)) ;
143.    N_zeta(7) = N_zeta(7) - 0.5* (N_zeta([illegible]) +N_zeta([illegible]) +...
        N_zeta(19)) ;
144.    N_zeta(8) = N_zeta(8) - 0.5* (N_zeta(11) +N_zeta(16) +...
        N_zeta(20)) ;
145.  return
146.
147.  function J = Jacobi( ie, xi, eta, zeta )
148.  %  计算 Jacobi 矩阵
149.  %  输入参数:
150.  %     ie  ----------  单元号
151.  %     xi,eta,zeta  ----  局部坐标
152.  %  返回值:
153.  %     J  ----------  在局部坐标(xi,eta,zeta)处的Jacobi矩阵
154.    global gNode gElement
155.    x = gNode(gElement(ie,1:20),1) ;
156.    y = gNode(gElement(ie,1:20),2) ;
157.    z = gNode(gElement(ie,1:20),3) ;
158.    [N_xi, N_eta, N_zeta] = N_xietazeta( ie, xi, eta, zeta ) ;
159.    x_xi   = N_xi   * x ;
160.    x_eta  = N_eta  * x ;
161.    x_zeta = N_zeta * x ;
162.
163.    y_xi   = N_xi   * y ;
164.    y_eta  = N_eta  * y ;
165.    y_zeta = N_zeta * y ;
166.
167.    z_xi   = N_xi   * z ;
168.    z_eta  = N_eta  * z ;
```



6.3.2 等效结点力

1. 体积力

设单位体积力是 $\boldsymbol{p}_v = [p_{vx} \quad p_{vy} \quad p_{vz}]^{\mathrm{T}}$，则移置到各结点上的等效结点力为

$$\boldsymbol{f}_{vi}^e = \begin{Bmatrix} f_{vxi} \\ f_{vyi} \\ f_{vzi} \end{Bmatrix} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} N_i \begin{Bmatrix} p_{vx} \\ p_{vy} \\ p_{vz} \end{Bmatrix} |\boldsymbol{J}| \mathrm{d}\xi \mathrm{d}\eta \mathrm{d}\zeta \quad (i = 1,2,\cdots,n) \qquad (6\text{-}100)$$

当体积力的形式比较简单，如为多项式时，则上式能积出显式，可以利用 ***MATLAB*** 的符号运算功能得到。但是一般可以利用高斯积分法，因为取合适的积分阶数时，可以得到多项式积分的精确值。设单元重力的方向为 z 的负方向，则采用 **3×3×3** 高斯积分法的单元自重的等效结点力为

$$f_{vzi} = -\rho g \sum_{m=1}^{3} \sum_{n=1}^{3} \sum_{k=1}^{3} W_m W_n W_k N_i(\xi_m, \xi_n, \xi_k) |\boldsymbol{J}(\xi_m, \xi_n, \xi_k)| \quad (i = 1,2,\cdots,n) \qquad (6\text{-}101)$$

式中 W_m 和 $\xi_m(m=1,2,3)$ 分别是高斯积分的权系数和高斯点。另外两个 x 和 y 方向的等效结点力分量为零。

2. 表面力

设单元的某边界 S 上作用表面力 $\boldsymbol{p}_s = [p_{sx} \quad p_{sy} \quad p_{sz}]^{\mathrm{T}}$，则在此面上各结点的等效结点力为

$$\boldsymbol{f}_{si}^e = \begin{Bmatrix} f_{sxi} \\ f_{syi} \\ f_{szi} \end{Bmatrix} = \iint_S N_i \begin{Bmatrix} p_{sx} \\ p_{sy} \\ p_{sz} \end{Bmatrix} \mathrm{d}S \qquad (6\text{-}102)$$

式中曲面积分是在单元上作用分布力 $\boldsymbol{p}_s$ 的某个边界面上进行的，例如，对于 $\zeta=1$ 的面上进行积分，按照数学分析[8]的公式有

$$dS = |s \times t|_{\zeta=1} d\xi d\eta \tag{6-103}$$

将它代入式(6-102),得到

$$f_{si} = \int_{-1}^{1}\int_{-1}^{1}(N_i p_s \, |s \times t|)_{\zeta=1} d\xi d\eta \quad (i \text{ 取 } S \text{ 上的结点}) \tag{6-104}$$

若单元的某个面上只作用着沿外法线方向荷载 σ,设 n 表示该曲面的外法线方向,则有

$$p_{sx} dS = \sigma \cos(n, x) dS = \sigma dy dz \quad (x, y, z) \tag{6-105}$$

于是式(6-102)可以写成

$$f_{sxi} = \iint N_i \sigma dy dz \quad (x, y, z) \quad (i \text{ 取 } S \text{ 上的结点}) \tag{6-106}$$

这就是将原来的第一类曲面积分化为第二类曲面积分。例如,对于 $\zeta = \pm 1$ 的面上,由坐标变换式(6-78),给出 y、z 和 ξ、η 之间的关系,因此 $dydz$ 可以用 $d\xi d\eta$ 来表示,通过计算和归纳,可以将式(6-106)写成

$$f_{si} = \begin{Bmatrix} f_{sxi} \\ f_{syi} \\ f_{szi} \end{Bmatrix} = \pm \int_{-1}^{1}\int_{-1}^{1}(\sigma N_i \, s \times t)_{\zeta=\pm 1} d\xi d\eta \quad (i \text{ 取 } S \text{ 上的结点}) \tag{6-107}$$

对于 $\xi = \pm 1$ 及 $\eta = \pm 1$ 的面上,相应的计算公式只须在上式右端对 ξ、η、ζ 和 s、t、v 同时进行轮换就可得到。

基于实际的考虑,式(6-104)和式(6-107)使用时并不方便,因为必须知道作用在单元的哪一个面来确定积分变量,还需要考虑正负符号。其实根据形函数的特性,即不在某一个面上的结点的形函数在这个面上值为零,因此表面力只对作用面上的结点有贡献。设单元的某一个曲面上作用有分布表面力,它由 $n(n=4\sim 8)$ 个空间结点组成,该曲面可以用参数方程写成

$$x = \sum_{i=1}^{n} N_i(\xi,\eta) x_i, \quad y = \sum_{i=1}^{n} N_i(\xi,\eta) y_i, \quad z = \sum_{i=1}^{n} N_i(\xi,\eta) z_i \tag{6-108}$$

式中 x_i,y_i 和 z_i 是 n 个结点在整体坐标系下的坐标值,而 $N_i(\xi,\eta)$ 就是 4~8 结点等参数单元的形函数,即由式(6-15)~式(6-22)定义,而不是 20 结点单元的形函数。这里要注意的是,这 n 个结点也必须如图 6-4 所示的 4~8 结点等参数单元的结点顺序排列。则表面力 $p_s = [p_{sx} \quad p_{sy} \quad p_{sz}]^T$ 在这 n 个结点上的等效结点力是

$$\begin{Bmatrix} f_{sxi} \\ f_{syi} \\ f_{szi} \end{Bmatrix} = \iint_S N_i(\xi,\eta) \begin{Bmatrix} p_{sx} \\ p_{sy} \\ p_{sz} \end{Bmatrix} \mathrm{d}S \qquad (i = 1,2,\cdots,n) \tag{6-109}$$

而分布法向力 σ 的等效结点力是

$$\begin{Bmatrix} f_{sxi} \\ f_{syi} \\ f_{szi} \end{Bmatrix} = \iint_S N_i(\xi,\eta)\sigma \begin{Bmatrix} \mathrm{d}y\mathrm{d}z \\ \mathrm{d}x\mathrm{d}z \\ \mathrm{d}x\mathrm{d}y \end{Bmatrix} \qquad (i = 1,2,\cdots,n) \tag{6-110}$$

式(6-109)是第一类曲面积分,而(6-110)是第二类曲面积分。当表面力的分布形式已知时,就能对这两类曲面积分进行计算。一般情况下,表面力也可以用形函数插值表示,即

$$\boldsymbol{p}_s = \sum_{i=1}^{8} N_i(\xi,\eta)\boldsymbol{p}_{si}, \quad \sigma = \sum_{i=1}^{8} N_i(\xi,\eta)\sigma_i \tag{6-111}$$

式中 $\boldsymbol{p}_{si} = [p_{sxi} \quad p_{syi} \quad p_{szi}]^{\mathrm{T}}$, $\sigma_i(i=1,2,\cdots,n)$分别是 n 个结点上的分布力值。则

$$\iint_S N_i(\xi,\eta)p_{sx}\mathrm{d}S = \int_{-1}^{1}\int_{-1}^{1} N_i(\xi,\eta)\Big[\sum_{j=1}^{8} N_j(\xi,\eta)p_{sxj}\Big]\sqrt{EG-F^2}\,\mathrm{d}\xi\mathrm{d}\eta$$

$$(x,y,z) \qquad (i = 1,2,\cdots,n) \tag{6-112}$$

式中

$$E = x_{,\xi}^2 + y_{,\xi}^2 + z_{,\xi}^2, \quad G = x_{,\eta}^2 + y_{,\eta}^2 + z_{,\eta}^2, \quad F = x_{,\xi}x_{,\eta} + y_{,\xi}y_{,\eta} + z_{,\xi}z_{,\eta} \tag{6-113}$$

而式(6-110)则化为

$$\begin{Bmatrix} f_{sxi} \\ f_{syi} \\ f_{szi} \end{Bmatrix} = \iint_S N_i(\xi,\eta)\sigma \begin{Bmatrix} \mathrm{d}y\mathrm{d}z \\ \mathrm{d}x\mathrm{d}z \\ \mathrm{d}x\mathrm{d}y \end{Bmatrix}$$

$$= \int_{-1}^{1}\int_{-1}^{1} N_i(\xi,\eta)\Big[\sum_{j=1}^{8} N_j\sigma_j\Big] \begin{Bmatrix} y_{,\xi}z_{,\eta} - y_{,\eta}z_{,\xi} \\ z_{,\xi}x_{,\eta} - z_{,\eta}x_{,\xi} \\ x_{,\xi}y_{,\eta} - x_{,\eta}y_{,\xi} \end{Bmatrix} \mathrm{d}\xi\mathrm{d}\eta$$

$$(i = 1,2,\cdots,n) \tag{6-114}$$

而整体坐标(x,y,z)对参考坐标 ξ、η 的导数为

$$x_{,\xi} = \sum_{i=1}^{n} N_{i,\xi}x_i, \quad x_{,\eta} = \sum_{i=1}^{n} N_{i,\eta}x_i \qquad (x,y,z) \tag{6-115}$$

现在,式(6-112)和式(6-114)中的被积函数全部被表示成积分变量 ξ 和 η 的

函数，利用高斯积分公式可容易地得到表面力的等效结点力。当被积函数是多项式时，可利用 MATLAB 的符号运算功能得到结果，但是结果过于复杂，因此还是采用数值积分为宜。

下面是计算分布压力等效结点力的程序段。变量 node 是定义压力作用面的结点号，pressure 是对应结点上的压力值。计算采用 3×3 的高斯积分法。

```
1.  function edf = EquivalentDistPressure( node, pressure )
2.  %  计算分布压力的等效结点力
3.  %  输入参数：
4.  %     node ---------- 结点号
5.  %     pressure ------- 与结点号对应的压力值
6.  %  返回值：
7.  %     edf ------------- 等效结点力向量
8.    global gNode
9.      x = gNode( node, 1 ) ;
10.     y = gNode( node, 2 ) ;
11.     z = gNode( node, 3 ) ;
12.     g = [ -0.774596669241483,                0.0,0.774596669241483];
13.     w = [  0.555555555555556,0.888888888888889,0.555555555555556];
14.     edf = zeros( 8*3, 1 ) ;
15.     for i=1:length(g)
16.        for j=1:length(g)
17.           xi = g(i) ;
18.           eta = g(j) ;
19.           N5 = ( eta - 1 ) * ( xi^2 - 1 ) /2 ;
20.           N6 = ( xi + 1 ) * ( 1 - eta^2 ) / 2 ;
21.           N7 = ( eta + 1 ) * ( 1 - xi^2 ) /2 ;
22.           N8 = ( xi - 1 ) * ( eta^2 - 1 ) / 2 ;
23.           N1 = ( 1 - xi ) * ( 1 - eta ) / 4 - 0.5 * ( N8 + N5 ) ;
24.           N2 = ( 1 + xi ) * ( 1 - eta ) / 4 - 0.5 * ( N5 + N6 ) ;
25.           N3 = ( 1 + xi ) * ( 1 + eta ) / 4 - 0.5 * ( N6 + N7 ) ;
26.           N4 = ( 1 - xi ) * ( 1 + eta ) / 4 - 0.5 * ( N7 + N8 ) ;
27.           N = [ N1 N2 N3 N4 N5 N6 N7 N8 ] ;
28.           N5_xi = (eta-1)*xi ;
29.           N6_xi = 1/2*(1-eta^2) ;
```

```
30.         N7_xi = -(eta+1)*xi;
31.         N8_xi = -1/2*(1-eta^2);
32.         N1_xi = -1/4*(1-eta) - 1/2*(N5_xi + N8_xi);
33.         N2_xi = 1/4*(1-eta) - 1/2*(N5_xi + N6_xi);
34.         N3_xi = 1/4*(eta+1) - 1/2*(N6_xi + N7_xi);
35.         N4_xi = -1/4*(eta+1) - 1/2*(N7_xi + N8_xi);
36.         N_xi = [N1_xi N2_xi N3_xi N4_xi N5_xi N6_xi N7_xi N8_xi];
37.         N5_eta = 1/2*(xi^2-1);
38.         N6_eta = -eta*(1+xi);
39.         N7_eta = 1/2*(1-xi^2);
40.         N8_eta = eta*(xi-1);
41.         N1_eta = -1/4*(1-xi) - 1/2*(N5_eta + N8_eta);
42.         N2_eta = -1/4*(1+xi) - 1/2*(N5_eta + N6_eta);
43.         N3_eta = 1/4*(1+xi) - 1/2*(N6_eta + N7_eta);
44.         N4_eta = 1/4*(1-xi) - 1/2*(N7_eta + N8_eta);
45.         N_eta = [N1_eta N2_eta N3_eta N4_eta ...
                    N5_eta N6_eta N7_eta N8_eta];
46.         x_xi = N_xi*x;
47.         y_xi = N_xi*y;
48.         z_xi = N_xi*z;
49.         x_eta = N_eta*x;
50.         y_eta = N_eta*y;
51.         z_eta = N_eta*z;
52.         px = y_xi*z_eta - y_eta*z_xi;
53.         py = z_xi*x_eta - z_eta*x_xi;
54.         pz = x_xi*y_eta - x_eta*y_xi;
55.         sig = N*[illegible];
56.         edf = edf + w(i)*w(j)*sig*[N1 ... [illegible]
                  N3*[illegible] ...
57.               N8*[illegible]];
58.     end
59. end
60. return
```

3.温度应力

如果考虑温度变化产生的初应变，则单元各结点上还应加上温度改变引起的等效结点力

$$
f_{Ti} = \begin{Bmatrix} f_{Txi} \\ f_{Tyi} \\ f_{Tzi} \end{Bmatrix} = \iiint_V B_i D\varepsilon_0 \, dxdydz
$$

$$
= \frac{E\alpha}{1-2\mu}\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} T \begin{Bmatrix} N_{i,x} \\ N_{i,y} \\ N_{i,z} \end{Bmatrix} |J| d\xi d\eta d\zeta
$$

$$
(i = 1, 2, \cdots, n) \qquad (6\text{-}116)
$$

计算应力的公式相应地修改为

$$
\sigma = DB\delta^{e} - D\varepsilon_0 \qquad (6\text{-}117)
$$

我们也可以构造更高阶的等参数单元，如 32 结点的等参数单元，它在每个棱边上有两个中间结点，在每单元中这两个结点分别安置在棱边长的三分之一处和三分之二处。实际使用时，划分单元应注意边棱的夹角不宜太锐或太钝，还应使平面 8 结点单元和空间 20 结点的中间上结点选在边棱的中点或其邻近处；对于平面 12 结点单元和空间 32 结点单元，则中间结点宜选在离角点的三分之一边长处为好。

在实际划分单元时，经常出现棱边全部为直线或大部分为直线的单元，对于一条直线，只须两个端点坐标就能确定，也就是说，不需要中间结点就能确定该直线，特别是直棱折面的六面体，只需 8 个角结点坐标，利用 8 结点等参变换形函数，就能确定单元的几何形状。因此，在确定位移模式和确定几何形状时，可以使用不同结点数的形函数，一般来说，只须把坐标变换式(6-78)改写成如下形式

$$
x = \sum_{i=1}^{m} N_i^{*} x_i, \quad y = \sum_{i=1}^{m} N_i^{*} y_i, \quad z = \sum_{i=1}^{m} N_i^{*} z_i \qquad (6\text{-}118)
$$

式中 m 是确定单元几何形状所用的结点数；N_i^{*} 是相应的形函数。

式(6-79)中的 n 是确定单元位移模式所用的结点，显然，不必要求 $m = n$。当 $m = n$ 时，称为等参数单元(isoparametric)；当 $m < n$ 时，称为亚参数单元(subparametric)；当 $m > n$ 时，称为超参数单元(superparametric)。利用亚参数单元，能提高计算效率。例如，对于直边折面的六面体，8 个角结点就可确定单元的几何形状，及 $m = 8$，坐标变换中的形函数 N_i^{*} 即是式(6-78)中的 N_i，于是涉及几何形状变换的计算，如 **Jacobi** 矩阵的计算量减少，其他有关的计算量也相应地减少，从而提高了效率。而超参数单元则可以在不增加单元数目的

情况下更精确地模拟复杂的结构形状。

对于六面体等参数单元的数值积分，Irons[8]曾提出过一个 14 点的积分公式，能达到 3×3×3 高斯积分同样的精度。该积分公式直接写成一次求和的形式，即

$$
\begin{aligned}
\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta,\zeta)\mathrm{d}\xi\mathrm{d}\eta\mathrm{d}\zeta = {} & B_0[f(-b,0,0)+f(b,0,0) \\
& +f(0,-b,0)+f(0,b,0)+f(0,0,-b)+f(0,0,b)] \\
& +C_0[f(-c,-c,-c)+f(c,c,c)+f(c,-c,-c)+f(-c,c,-c) \\
& +f(-c,-c,c)+f(c,c,-c)+f(-c,c,c)+f(c,-c,c)]
\end{aligned}
\tag{6-119}
$$

式中

$$
B_0=320/361, C_0=121/361, b=\sqrt{19}/\sqrt{30}, c=\sqrt{19}/\sqrt{33} \tag{6-120}
$$

6.4 算例

6.4.1 受内压的旋转厚壁圆筒

设厚壁圆筒的内径为 10cm，外径为 20cm，承受内压 120MPa，并以角速度 $\omega=2094\mathrm{rad/s}$ 绕中心轴旋转。材料的弹性模量 $E=210\mathrm{GPa}$，泊松比 $\mu=0.3$，密度 $\rho=7800\mathrm{kg/m^3}$。现计算该圆筒的位移和应力分布。

根据问题的对称性，可以取夹角等于 10°的两个径向剖面从圆筒中划出一块进行有限元分析，单元划分如图 6-7 所示。这里要特别强调的是边界条件

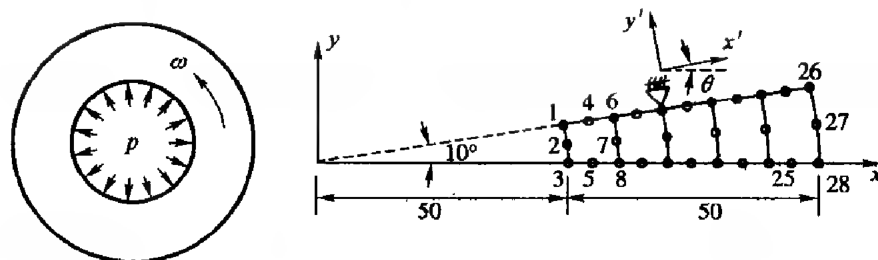


图 6-7 受内压的旋转圆筒及其有限元模型

的处理。根据对称性，在 $y=0$ 的边界上 y 方向的位移为零，在另一条半径边界上，是它的法向位移为零，即所谓的"斜支撑"。如图 6-7 所示，定义局部坐

标系 $x'y'$，它们与整体坐标系 xy 的夹角为 $\theta = 10°$。设 u' 和 v' 表示在局部坐标系 $x'y'$ 下的位移分量，则这条边界上的约束条件为 $v' = 0$。为了处理这种斜约束，可以把定义这种斜约束的结点位移进行坐标变换，这与第 3 章中处理杆件系统的局部坐标与整体坐标进行变换的过程非常相似。不同在于杆件系统中对所有结点进行坐标变换，而这里仅对定义斜约束的结点进行坐标变换。设在结点 i 处定义有斜约束，则可设整体坐标系下和部分局部坐标下的结点位移列阵

$$\boldsymbol{\delta} = [\boldsymbol{\delta}_1^{\mathrm{T}} \quad \boldsymbol{\delta}_2^{\mathrm{T}} \quad \cdots \quad \boldsymbol{\delta}_i^{\mathrm{T}} \quad \cdots \quad \boldsymbol{\delta}_n^{\mathrm{T}}]^{\mathrm{T}} \tag{6-121}$$

$$\boldsymbol{\delta}' = [\boldsymbol{\delta}_1^{\mathrm{T}} \quad \boldsymbol{\delta}_2^{\mathrm{T}} \quad \cdots \quad \boldsymbol{\delta}_i'^{\mathrm{T}} \quad \cdots \quad \boldsymbol{\delta}_n^{\mathrm{T}}]^{\mathrm{T}} \tag{6-122}$$

式中

$$\boldsymbol{\delta}_j = [u_j \quad v_j]^{\mathrm{T}} (j = 1, 2, \cdots, n), \quad \boldsymbol{\delta}'_i = [u'_i \quad v'_i]^{\mathrm{T}} \tag{6-123}$$

则有下列转换关系成立

$$\boldsymbol{\delta}' = \boldsymbol{T}\boldsymbol{\delta} \tag{6-124}$$

式中

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \cdots & \boldsymbol{0} & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{T}_i & \cdots & \boldsymbol{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \cdots & \boldsymbol{0} & \cdots & \boldsymbol{I} \end{bmatrix} \begin{matrix} \\ \\ \\ \text{第 } i \text{ 行} \\ \\ \\ \end{matrix} \quad (\text{第 } i \text{ 列}) \qquad \boldsymbol{T}_i = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ -\sin\theta_i & \cos\theta_i \end{bmatrix} \tag{6-125}$$

而 θ_i 是第 i 个结点斜约束的倾斜角度，则可以用第 3 章杆件系统中同样的方法，把原来的有限元方程化成

$$\boldsymbol{K}'\boldsymbol{\delta}' = \boldsymbol{f}' \tag{6-126}$$

式中

$$\boldsymbol{K}' = \boldsymbol{T}\boldsymbol{K}\boldsymbol{T}^{\mathrm{T}}, \boldsymbol{f}' = \boldsymbol{T}\boldsymbol{f} \tag{6-127}$$

由于矩阵转换 $\boldsymbol{T}$ 是分块意义上的对角阵，上式可以简写为

$$
\boldsymbol{K}' = \begin{bmatrix}
\boldsymbol{K}_{11} & \boldsymbol{K}_{12} & \cdots & \boldsymbol{K}_{1i}\boldsymbol{T}_i^{\mathrm{T}} & \cdots & \boldsymbol{K}_{1n} \\
\boldsymbol{K}_{21} & \boldsymbol{K}_{22} & \cdots & \boldsymbol{K}_{2i}\boldsymbol{T}_i^{\mathrm{T}} & \cdots & \boldsymbol{K}_{2n} \\
\vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\
\boldsymbol{T}_i\boldsymbol{K}_{i1} & \boldsymbol{T}_i\boldsymbol{K}_{i2} & \cdots & \boldsymbol{T}_i\boldsymbol{K}_{ii}\boldsymbol{T}_i^{\mathrm{T}} & \cdots & \boldsymbol{T}_i\boldsymbol{K}_{in} \\
\vdots & \vdots & \cdots & \vdots & \ddots & \vdots \\
\boldsymbol{K}_{n1} & \boldsymbol{K}_{n2} & \cdots & \boldsymbol{K}_{ni}\boldsymbol{T}_i^{\mathrm{T}} & \cdots & \boldsymbol{K}_{nn}
\end{bmatrix}
\quad
\boldsymbol{f}' = \begin{Bmatrix}
\boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \vdots \\ \boldsymbol{T}_i\boldsymbol{f}_i \\ \vdots \\ \boldsymbol{f}_n
\end{Bmatrix}
\tag{6-128}
$$

式中 $\boldsymbol{K}_{ij}$和 $\boldsymbol{f}_i$分别是原刚度矩阵和结点力列阵的子矩阵。这样，在求解方程（6-126）时，第 i 个结点在斜约束方向的位移分量已经显式地出现在位移列阵 $\boldsymbol{\delta}'$中，因此对它们的约束处理完全可以按照普通的边界条件处理。最后，要注意的是，如果要知道在整体坐标系下的位移，或者计算应力应变时，应使用式（6-124）把位移转换到整体坐标系下。

下面是处理斜约束的程序段。

```
1.  % 处理斜约束边界条件
2.  [bc3_number,dummy] = size(gBC3) ;
3.  for ibc=1:1:bc3_number
4.      n = gBC3( ibc, 1 ) ;
5.      theta = gBC3( ibc, 4 ) ;
6.      c = cos(theta) ;
7.      s = sin(theta) ;
8.      T = [ c s
9.           -s c ] ;
10.         gK((n-1)*2+1:(n-1)*2+2,:) = T* gK((n-1)*2+1:(n-1)*2+2,:) ;
11. [illegible]
12.         f((n-1)*2+1:(n-1)*2+2,:) = T* f((n-1)*2+1:(n-1)*2+2,:) ;
13.         gBC1 = [gBC1; gBC3(ibc,1:3)] ;
14. end
```

这里变量 gBC3 中是斜约束边界条件，变量 gBC1 中是普通边界条件，因此把刚度矩阵 gK 和结点力向量 f 变换到斜约束的坐标下后，直接把斜约束条件放到普通边界条件中去即可（上面程序第 13 行）。下面的程序段把结点位移转换到整体坐标系下。

```
15. % [illegible]
16.     [illegible] = gDelta ;
17.     for ibc=1:[illegible]
18.         n = gBC3( ibc, 1 ) ;
19.         theta = gBC3( ibc, 2 ) ;
20.         c = cos(theta) ;
21.         s = sin(theta) ;
22.         T = [ c s
23.              -s c ] ;
24.         gDelta( (n-1)*2+1:n*2, : ) = [illegible](T)*...
                                        gDelta( (n-1)*2+1:n*2, : ) ;
25.     end
```

表 6-2 给出了圆筒的径向位移，并列出了解析解[10,11]。从表中可以发现只用 5 个单元就能得到非常精确的结果。表 6-3 和表 6-4 给出了径向应力和周向应力，计算方法是直接用结点的局部坐标代入计算，并取单元平均。从表中可以发现应力的精度比位移的精度要差，尤其在自由表面应力为零处。

圆筒的径向位移(mm) 表 6-2

结点号	坐标 x (mm)	承受内压		承受离心力	
		有限元	解析解[10]	有限元	解析解[11]
3	50	0.059001	0.059000	0.074282	0.074282
5	55	0.054973	0.054973	0.072395	0.072395
8	60	0.051734	0.051733	0.071042	0.071042
10	65	0.049100	0.049100	0.070049	0.070049
13	70	0.046943	0.046943	0.069285	0.069285
15	75	0.045167	0.045167	0.068654	0.068654
18	80	0.043700	0.043700	0.068076	0.068076
20	85	0.042488	0.042488	0.067490	0.067490
23	90	0.041489	0.041489	0.066843	0.066844
25	95	0.040668	0.040668	0.066092	0.066092
28	100	0.040000	0.040000	0.065197	0.065197

圆管的径向应力(MPa)　　表 6-3

结点号	坐标 x (mm)	承受内压		承受离心力	
		有限元	解析解[10]	有限元	解析解[11]
3	50	-116.63	-120.00	3.21	0.00
5	55	-93.81	-92.23	15.58	17.08
8	60	-68.79	-71.11	29.85	27.59
10	65	-55.48	-54.67	32.44	33.26
13	70	-40.39	-41.63	36.55	35.24
15	75	-31.57	-31.11	33.78	34.29
18	80	-21.78	-22.50	31.80	30.95
20	85	-15.64	-15.36	25.25	25.60
23	90	-8.94	-9.38	19.14	18.53
25	95	-4.50	-4.32	9.68	9.95
28	100	0.35	0.00	0.49	0.00

圆管的周向应力(MPa)　　表 6-4

结点号	坐标 x (mm)	承受内压		承受离心力	
		有限元	解析解[10]	有限元	解析解[11]
3	50	201.00	200.00	298.08	297.13
5	55	171.76	172.23	267.93	268.38
8	60	151.80	151.11	245.76	245.08
10	65	134.43	134.67	225.27	225.51
13	70	122.00	121.63	208.92	208.53
15	75	110.97	111.11	193.21	193.36
18	80	102.71	102.50	179.73	179.48
20	85	95.28	95.36	166.38	166.48
23	90	89.52	89.38	154.28	154.10
25	95	84.27	84.32	142.04	142.12
28	100	80.10	80.00	130.54	130.39

6.4.2 端部受剪力的圆截面悬臂梁

设有一圆形截面悬臂梁，在自由端圆心处作用一集中剪力 $Q=4000\text{kN}$，如图 6-8 所示，试计算梁的变形和应力。

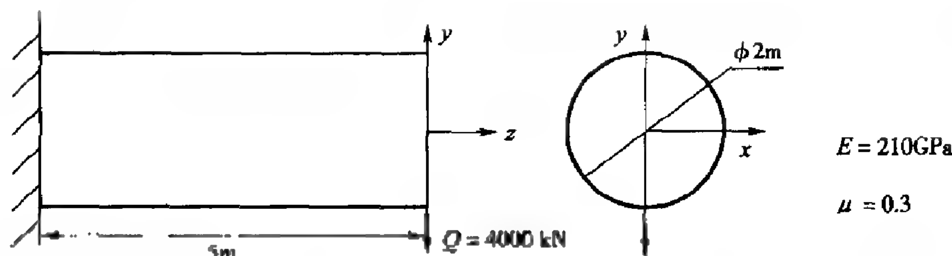


图 6-8 端部受剪力下圆截面悬臂梁

设圆形截面直径为 1m，长度 5m，弹性模量 $E=210\text{GPa}$，泊松比 $\mu=0.3$。根据对称性，取 1/4 计算，采用 20 结点空间单元，网格如图 6-9 所示。根据对称性，在 $x=0$ 的平面上，x 方向的位移为零，而在 $y=0$ 的平面上，z 方向的位移为零。在固定端，3 个方向的位移均为零。程序中采用 14 点的 Irons 积分公式计算刚度矩阵，结点应力采用绕结点平均法。

表 6-5 给出了横截面（该截面的轴向坐标 $z=-2.3077$，即距离自由端 2.3077m）上轴向正应力和竖向剪应力的分布情况，并与解析解[12]的结果做了比较。从表上可以看出，有限元的计算精度还是很高的，正应力的精度要高于剪应力，这是因为正应力沿梁高度是线性分布的，而剪应力沿梁宽度是二次抛物线分布的，后者要求更高阶的位移插值函数。

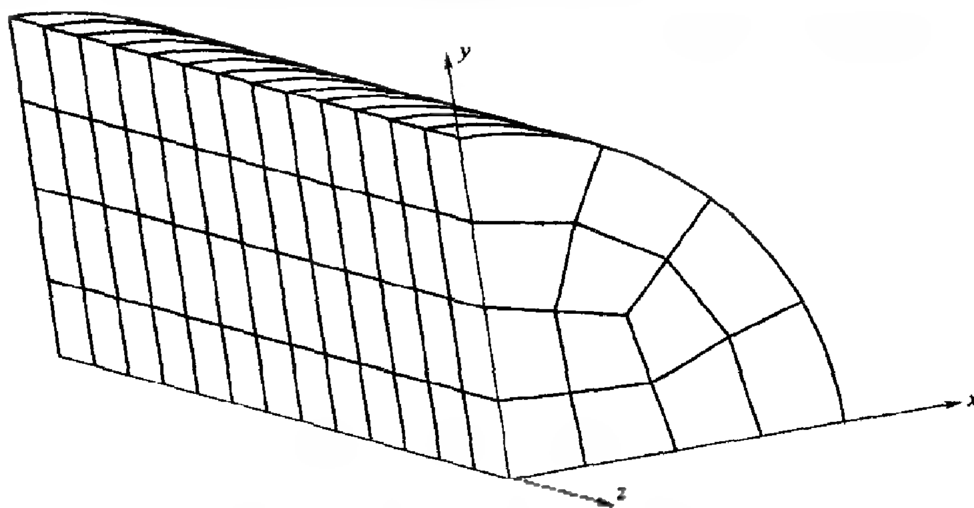


图 6-9 端部受剪力的圆截面悬臂梁的有限元网格

表 6-5 轴向正应力和竖向剪应力的分布

y 坐标 (m)	轴向正应力(MPa) 有限元	轴向正应力(MPa) 解析解[12]	x 坐标 (m)	竖向剪应力(MPa) 有限元	竖向剪应力(MPa) 解析解[12]
1.000	11.66	11.75	0.000	-1.81	-1.76
0.875	10.25	10.28	0.125	-1.80	-1.76
0.750	8.83	8.81	0.250	-1.79	-1.75
0.625	7.33	7.35	0.375	-1.77	-1.74
0.500	5.88	5.88	0.500	-1.75	-1.71
0.375	4.41	4.41	0.625	-1.72	-1.69
0.250	2.94	2.94	0.750	-1.69	-1.65
0.125	1.47	1.47	0.875	-1.65	-1.61
0.000	0.01	0.00	1.000	-1.59	-1.57

参考文献

[1] Irons B. M.. Engineering application of numerical integration in stiffness method. Journal of the American Institute of Aeronautics and Astronautics. ,1966.4(11): 2035-2037

[2] Ergatoudis, I., Irons, B. M., Zienkiewicz O. C.. Curved isoparametric, quadrilateral element for finite element analysis. International Journal of Solids and Structures,1968, 4: 31-42.

[3] Zienkiewicz, O. C.. The Finite Element Method. 3rd ed.. London:McGraw-Hill,1977.

[4] Bathe K. J., Wilson E. L.. Numerical Methods in Finite Element Analysis Englewood Cliffs, NJ. :Prentice-Hall,1976.

[5] Cook R. D. . Concepts and Applications of Finite Element Analysis. John Wiley & Sons Inc. ,2002.

[6] Stroud A. H., Secrest D.. Gaussian Quadrature Formulas. Englewood Cliffs, NJ. :Prentice Hall,1966.

[7] 王勖成. 有限单元法. 北京:清华大学出版社,2003.

[8] Irons B. M.. Quadrature rules for brick based finite elements. International Journal for Numerical Methods in Engineering, 1971,3(2):293-294.

[9] R. 柯朗, F. 约翰. 微积分和数学分析引论(第二卷). 北京: 科学出版社, 1989

[10] 徐芝纶. 弹性力学(上册). 2版. 北京: 人民教育出版社

[11] Timoshenko S. P. Goodier J. N. 1970 Theory of elasticity. New York: McGraw-Hill

[12] 谢贻权, 林钟祥, 丁皓江. 弹性力学. 杭州: 浙江大学出版社, 1988.

第7章 板壳单元

板和壳是指厚度比其他尺寸要小得多的平面或曲面构件。板和壳在工程中应用广泛，由于它们的这种几何特点，前面所述的三维单元并不适合用来分析它们的变形，这是因为三维单元要求在3个方向的尺寸应尽量接近，否则求解精度会由于“剪切自锁”(**shear locking**)或系统矩阵病态而大大降低，甚至得到错误的结果，所以必须采用很细密的网格来适应板和壳的几何特征。但是采用这种方法将导致有限元模型的自由度“疯狂”增长，花费大量的计算和前后处理时间。因此开发适合于板壳结构的专用单元是十分必要的，事实上，目前已经有各种各样的板壳单元出现[1-5]。

仿照根据梁理论建立梁单元的思路，自然想到根据板理论建立板单元。这里讨论两种板理论，一种是薄板理论[6]，也被称为 **Kirchhoff** 板理论，该理论忽略板的横向剪切变形；另一种是 **Mindlin** 板理论[7]，该理论考虑板的横向剪切变形的影响，适合于板的厚跨比较大的情形，后者也常被称为 **Reissner** 板理论[8]或中厚板理论。根据这两种理论可以建立不同的板单元。

7.1 弹性板的弯曲

板的详细理论可以参见文献[6-8]，为了便于阅读，这里作简单叙述。设板的中面在 xy 平面上，即 $z=0$ 表示板的中面，如图7-1a)所示。在板理论中，一般假设板的中面是一中性面，也就是在没有面内力时，中面上的3个应变 $\varepsilon_x=\varepsilon_y=\gamma_{xy}=0$ 。另一个基本假设即为所谓的直法线假定：变形前垂直于中面的法线变形后

仍然保持直线，但是不一定仍然垂直于变形后的中面，如图 7-1b)所示。这条直线在绕 y 和 x 轴的转角分别为 ψ_x 和 ψ_y，则距离中面距离为 z 的任意点的位移和应变分别是

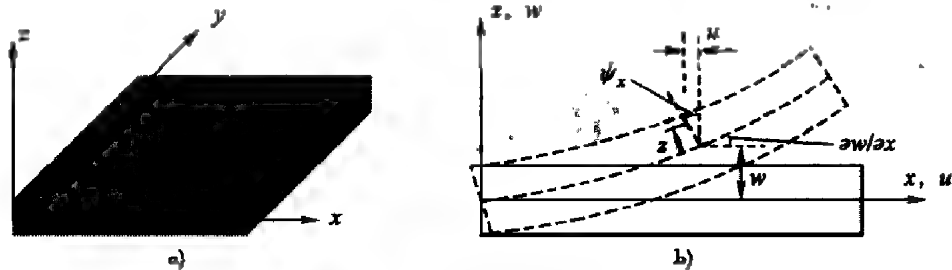


图 7-1　板及其变形

$$u = -z\psi_x \quad v = -z\psi_y \quad \varepsilon_x = -z\psi_{x},_x \quad \varepsilon_y = -z\psi_{y},_y$$
$$\gamma_{xy} = -z(\psi_{x},_y + \psi_{y},_x) \quad \gamma_{yz} = w,_y - \psi_y \quad \gamma_{xz} = w,_x - \psi_x \tag{7-1}$$

这里 w 是板的横向挠度，假设它沿板的厚度方向不变，即 $\varepsilon_z = 0$。上式是 **Mindlin** 板理论的基本假定，如果假定变形后的法线仍然是变形后中面的法线，即 $w,_x = \psi_x$ 和 $w,_y = \psi_y$，则式(7-1)中的两个横向剪切应变 γ_{yz} 和 γ_{xz} 为零，这就退化为 **Kirchhoff** 板理论。当板足够薄时，用 **Kirchhoff** 板理论能得到符合实际的结果。

在截面上的应力分别如图 7-2 所示。在板理论中经常用内力，即弯矩和剪力来表示，它们与应力之间的关系为

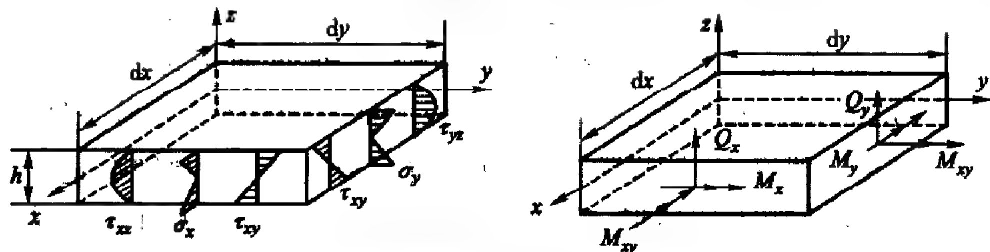


图 7-2　板横截面内的应力分布和内力

$$M_x = \int_{-h/2}^{-h/2} \sigma_x z\mathrm{d}z, \quad M_y = \int_{-h/2}^{-h/2} \sigma_y z\mathrm{d}z, \quad M_{xy} = \int_{-h/2}^{-h/2} \tau_{xy} z\mathrm{d}z,$$
$$Q_x = \int_{-h/2}^{-h/2} \tau_{xz}\mathrm{d}z, \quad Q_y = \int_{-h/2}^{-h/2} \tau_{yz}\mathrm{d}z \tag{7-2}$$

式中 M_x、M_y 是弯矩，M_{xy} 是扭矩，Q_x 和 Q_y 是剪力。它们的正方向如图 7-2 所示。如果直法线假设成立，那么应力 σ_x、σ_y 和 τ_{xy} 沿厚度方向线性变化，则材料力学的公式 $\sigma = Mz/I$ 同样成立。横向剪应力 τ_{xz} 和 τ_{yz} 跟面内应力 σ_x、σ_y 和 τ_{xy} 相比一般小很多。对于均匀材料，横向剪应力沿厚度按抛物线变化，在

中面处最大，在上下两个表面为零。另一个正应力 σ_z 跟面内应力相比也是小量，可以忽略。这样，对于线弹性材料，板内的应力应变关系为

$$\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{Bmatrix}=\frac{E}{1-\mu^2}\begin{bmatrix}1 & \mu & 0\\ \mu & 1 & 0\\ 0 & 0 & (1-\mu)/2\end{bmatrix}\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix}=D\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix} \tag{7-3}$$

很容易发现板的弹性矩阵 D 跟平面应力状态是一致的。

7.1.1 Kirchhoff 板理论

由于忽略横向剪切变形，即 $w_{,x}=\psi_x$ 和 $w_{,y}=\psi_y$，因此板内所有的力学量都能用挠度 w 表示。比如，应变可以表示为挠度的曲率，即

$$\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix}=-z\begin{Bmatrix}w_{,xx}\\ w_{,yy}\\ 2w_{,xy}\end{Bmatrix} \tag{7-4}$$

把上式代入应力应变关系（7-3），然后两边乘 z 并沿板厚度方向积分，即得到弯矩与挠度的关系为

$$\begin{Bmatrix}M_x\\ M_y\\ M_{xy}\end{Bmatrix}=-D\begin{bmatrix}1 & \mu & 0\\ \mu & 1 & 0\\ 0 & 0 & (1-\mu)/2\end{bmatrix}\begin{Bmatrix}w_{,xx}\\ w_{,yy}\\ 2w_{,xy}\end{Bmatrix} \tag{7-5}$$

式中抗弯刚度 D 为

$$D=\frac{Eh^3}{12(1-\mu^2)} \tag{7-6}$$

它与梁的抗弯刚度 EI 类似。

因此，Kirchhoff 板的变形和应力状态完全可以由中面的挠度 $w(x,y)$ 来描述。

7.1.2 Mindlin 板理论

根据 Mindlin 板理论的假设，中面法线在变形后不再垂直于中面，因此必须采用 3 个位移分量 w、ψ_x 和 ψ_y 来描述板内的变形，即

$$\begin{Bmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{Bmatrix}=-z\begin{Bmatrix}\psi_{x,x}\\ \psi_{y,y}\\ \psi_{x,y}+\psi_{y,x}\end{Bmatrix} \tag{7-7}$$

则内力与位移的关系是

$$\begin{Bmatrix} M_x \\ M_y \\ M_{xy} \end{Bmatrix} = -D\begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & (1-\mu)/2 \end{bmatrix}\begin{Bmatrix} \psi_{x,x} \\ \psi_{y,y} \\ \psi_{x,y}+\psi_{y,x} \end{Bmatrix} \tag{7-8}$$

另外为了修正横向剪应力沿板厚均匀分布导致的误差，引入了所谓剪切修正因子 κ 来修正剪力，即

$$Q_x = \kappa Gh(w_{,x}-\psi_x),\quad Q_y = \kappa Gh(w_{,y}-\psi_y) \tag{7-9}$$

因此，由于考虑了横向剪切变形，Mindlin 板理论包含了 3 个互相耦合的中面挠度和中面法线的两个转角位移。这样，Mindlin 板理论看上去比 Kirchhoff 板理论更为复杂，但是在构造有限单元时，有时反而比 Kirchhoff 板理论更容易做到位移协调。当然，这是以克服当板的厚跨比很小时出现的“剪切自锁”现象为代价的。

应用有限元法求解板弯曲问题时，用一些离散的板单元代替原来连续的结构。每个结点有 3 个广义位移分量，即挠度 w、绕 x 轴的转角 θ_x 和绕 y 轴的转角 θ_y。挠度 w 的正方向跟 z 轴一致，转角则以按右手螺旋法则标出的矢量沿坐标轴正向为正。结点的 3 个位移与中面的挠度及中面法线的旋转相对应，对于 Kirchhoff 板理论，为

$$\begin{Bmatrix} w \\ \theta_x \\ \theta_y \end{Bmatrix} = \begin{Bmatrix} w \\ w_{,y} \\ -w_{,x} \end{Bmatrix} \tag{7-10}$$

而对于 Mindlin 板理论，则有

$$\begin{Bmatrix} w \\ \theta_x \\ \theta_y \end{Bmatrix} = \begin{Bmatrix} w \\ \psi_y \\ -\psi_x \end{Bmatrix} \tag{7-11}$$

在这里要特别注意，在右手坐标系中，中面的倾角 $w_{,x}$（或 ψ_x）和 $w_{,y}$（或 ψ_y）分别对应绕 y 轴和 x 轴的转角位移，而且 $w_{,x}$（或 ψ_x）与 θ_y 的符号相反，如图 7-3所示。

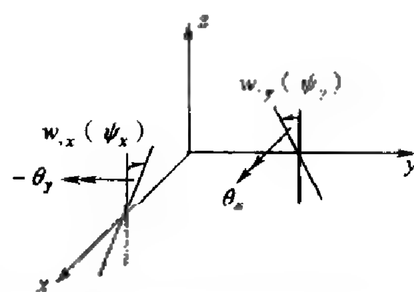


图 7-3　板理论的转角位移与有限元转角位移分量的关系

7.2　矩形薄板单元

矩形单元是薄板单元中比较简单的一种。平板用矩形单元离散，如图 7-4

所示。每一个结点有 3 个位移分量,即挠度 w、转角 θ_x 和 θ_y。则结点 i 的位移列阵为

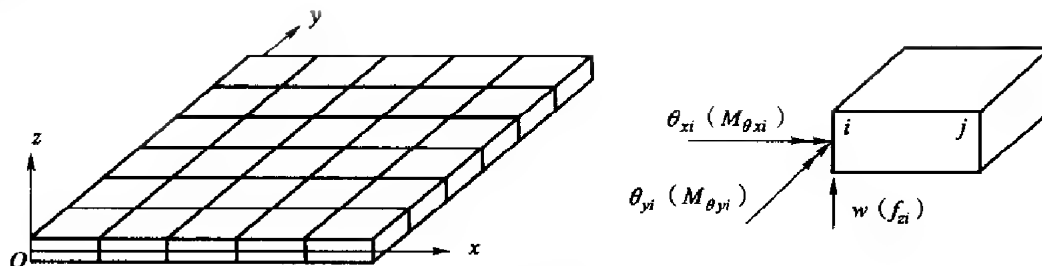


图 7-4　平板划分成矩形单元

$$\boldsymbol{\delta}_i = \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \end{Bmatrix} = \begin{Bmatrix} w_i \\ w_{,yi} \\ -w_{,xi} \end{Bmatrix} \tag{7-12}$$

与它相对应的结点力列阵是

$$\boldsymbol{f}_i = \begin{Bmatrix} f_{zi} \\ M_{\theta xi} \\ M_{\theta yi} \end{Bmatrix} \tag{7-13}$$

7.2.1　位移模式

根据矩形单元的形状特点，引入一个自然坐标系 $O\xi\eta$ 来研究单元特性是十分合适的。取矩形的中心为自然坐标系的原点，ξ 轴与 x 轴平行，η 轴与 y 轴平行，正方向也与 xy 轴一致，即与平面问题矩形单元的类似（图 4-10）。设单元的长度和宽度分别为 $2a$ 和 $2b$，那么 $\xi\eta$ 与 xy 的转换关系跟平面问题矩形单元也一样，即如式(4-83)所示。由于每个结点有三个位移分量，一个 4 结点的矩形单元共有 12 个结点位移分量，因此可选取含有 12 个参数的多项式作为位移模式，即

$$\begin{aligned} w = {} & \alpha_1 + \alpha_2\xi + \alpha_3\eta + \alpha_4\xi^2 + \alpha_5\xi\eta + \alpha_6\eta^2 + \alpha_7\xi^3 \\ & + \alpha_8\xi^2\eta + \alpha_9\xi\eta^2 + \alpha_{10}\eta^3 + \alpha_{11}\xi^3\eta + \alpha_{12}\xi\eta^3 \end{aligned} \tag{7-14}$$

最后两项的选取使挠度 w 在单元边界有三次的形式。按照上式可以得出转角为

$$\theta_x = \frac{\partial w}{\partial y} = \frac{\partial w}{b\partial\eta} = \frac{1}{b}(\alpha_3 + \alpha_5\xi + 2\alpha_6\eta$$

$$+\alpha_8\xi^2+2\alpha_9\xi\eta+3\alpha_{10}\eta^2+\alpha_{11}\xi^3+3\alpha_{12}\xi\eta^2) \tag{7-15}$$

$$\theta_y=-\frac{\partial w}{\partial x}=-\frac{\partial w}{a\partial\xi}=-\frac{1}{a}(\alpha_2+2\alpha_4\xi+\alpha_5\eta+3\alpha_7\xi^2+2\alpha_8\xi\eta$$

$$+\alpha_9\eta^2+3\alpha_{11}\xi^2\eta+\alpha_{12}\eta^3) \tag{7-16}$$

将矩形单元的 4 个结点坐标(ξ_i,η_i)和结点位移($w_i,\theta_{xi},\theta_{yi}$)分别代入式(7-14)~式(7-16),就可以得到关于这 12 个参数的联立方程组。从中解出 α_1 至 α_{12},再代入式(7-14),经整理后,我们得到

$$w=\sum_{i=1}^{4}(N_iw_i+N_{xi}\theta_{xi}+N_{yi}\theta_{yi})=\sum_{i=1}^{4}\boldsymbol{N}_i\boldsymbol{\delta}_i=\boldsymbol{N}\boldsymbol{\delta}^e \tag{7-17}$$

$$\boldsymbol{N}=[\boldsymbol{N}_1\quad\boldsymbol{N}_2\quad\boldsymbol{N}_3\quad\boldsymbol{N}_4] \tag{7-18}$$

$$\boldsymbol{\delta}^e=[\boldsymbol{\delta}_1^{\mathrm T}\quad\boldsymbol{\delta}_2^{\mathrm T}\quad\boldsymbol{\delta}_3^{\mathrm T}\quad\boldsymbol{\delta}_4^{\mathrm T}]^{\mathrm T} \tag{7-19}$$

$$\boldsymbol{N}_i=[N_i\quad N_{xi}\quad N_{yi}]\quad(i=1,2,3,4) \tag{7-20}$$

式中形函数是

$$\begin{aligned}
&N_i=(1+\xi_i\xi)(1+\eta_i\eta)(2+\xi_i\xi+\eta_i\eta-\xi^2-\eta^2)/8\\
&N_{xi}=-b\eta_i(1+\xi_i\xi)(1+\eta_i\eta)(1-\eta^2)/8 \qquad (i=1,2,3,4)\\
&N_{yi}=a\xi_i(1+\xi_i\xi)(1+\eta_i\eta)(1-\xi^2)/8
\end{aligned} \tag{7-21}$$

式(7-14)中的前三项反映了薄板单元的 3 个刚体位移。另外由式(7-4)看到,板内各点的应变完全由挠度 w 的 3 个二阶导数所决定,因此,式(7-14)中的第 4、5、6 三个二次项反映了这个常应变状态(或称常曲率状态)。所以这个矩形单元是完备的。

由于应变是位移的二阶导数,因此如果要得到一个协调的单元还要求在单元的交界面上有挠度 w 及其一阶导数的连续性,这个要求经常使问题复杂化。例如,在图 7-4 右边矩形单元的一条边 ij 上,y(或 η)是常数,挠度 w 是 x(或 ξ)的三次多项式,而 4 个结点位移 w_i w_j、θ_{yi} 和 θ_{yj} 正好可以确定它,因此在单元的一条边 ij 上,挠度和 θ_y 是连续的。但是 θ_x 则不能保证连续,这是因为 $\theta_x=\partial w/\partial y$ 也是 x(或 ξ)的三次多项式,而只有两个结点位移 θ_{xi} 和 θ_{xj} 可以部分约束它,所以无法确定一条三次曲线。因此,矩形单元在单元边界上挠度 w 和挠度沿边界方向的偏导数 $\partial w/\partial s$ 的值在单元交界线之间是连续的,而对于 $\partial w/\partial n$ 却并不能保证连续(s 表示交界线切线方向,而 n 表示交界线法线方向)。因此,矩形薄板单元是非协调单元。

7.2.2 刚度矩阵

将式(7-17)代入几何方程(7-4)，可以将单元应变用结点位移列阵表示为

$$\boldsymbol{\varepsilon} = \boldsymbol{B}\boldsymbol{\delta}^e = z[\boldsymbol{B}_1 \ \boldsymbol{B}_2 \ \boldsymbol{B}_3 \ \boldsymbol{B}_4]\boldsymbol{\delta}^e \tag{7-22}$$

式中

$$\boldsymbol{B}_i = -\begin{Bmatrix} N_{i,xx} \\ N_{i,yy} \\ 2N_{i,xy} \end{Bmatrix} = -\begin{Bmatrix} N_{i,\xi\xi}/a^2 \\ N_{i,\eta\eta}/b^2 \\ 2N_{i,\xi\eta}/(ab) \end{Bmatrix} = -\frac{1}{ab}\begin{Bmatrix} bN_{i,\xi\xi}/a \\ aN_{i,\eta\eta}/b \\ 2N_{i,\xi\eta} \end{Bmatrix} \quad (i=1,2,3,4) \tag{7-23}$$

符号 $N_{i,xx}$、$N_{i,\xi\xi}$等分别表示$\partial^2 N_i/\partial x^2$、$\partial^2 N_i/\partial \xi^2$等。按照式(7-21)容易得到

$$-\frac{b}{a}N_{i,\xi\xi} = \frac{1}{4}\left[\frac{3b}{a}\xi_i\xi(1+\eta_i\eta) \quad 0 \quad b\xi_i(1+3\xi_i\xi)(1+\eta_i\eta)\right] \tag{7-24}$$

$$-\frac{b}{a}N_{i,\eta\eta} = \frac{1}{4}\left[\frac{3a}{b}\eta_i\eta(1+\xi_i\xi) \quad -a\eta_i(1+\xi_i\xi)(1+3\eta_i\eta) \quad 0\right] \tag{7-25}$$

$$-2N_{i,\xi\eta} = \frac{1}{4}\left[\xi_i\eta_i(3\xi^2+3\eta^2-4) \quad -b\xi_i(3\eta^2+2\eta_i\eta-1) \quad a\eta_i(3\xi^2+2\xi_i\xi-1)\right] \tag{7-26}$$

于是单元刚度矩阵可以写成分块形式，其子矩阵的公式是

$$\boldsymbol{K}_{ij} = \iiint z^2 \boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}_j \mathrm{d}x\mathrm{d}y\mathrm{d}z = \frac{h^3}{12}\int_{-1}^{1}\int_{-1}^{1}\boldsymbol{B}_i^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}_j ab\mathrm{d}\xi\mathrm{d}\eta \quad (i,j=1,2,3,4) \tag{7-27}$$

把应变矩阵式(7-23)和式(7-3)中的弹性矩阵 $\boldsymbol{D}$ 代入上式，并完成积分运算得到矩形薄板单元显式的单元刚度子矩阵

$$\boldsymbol{K}_{ij} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{7-28}$$

式中 9 个元素如下

$$a_{11} = 3H\left[15\left(\frac{b^2}{a^2}\xi_0 + \frac{a^2}{b^2}\eta_0\right) + \left(14-4\mu+5\frac{b^2}{a^2}+5\frac{a^2}{b^2}\right)\xi_0\eta_0\right] \tag{7-29}$$

$$a_{12} = -3Hb\left[\left(2+3\mu+5\frac{a^2}{b^2}\right)\xi_0\eta_i + 15\frac{a^2}{b^2}\eta_i + 5\mu\xi_0\eta_j\right] \tag{7-30}$$

$$a_{13} = 3Ha\left[\left(2 + 3\mu + 5\frac{b^2}{a^2}\right)\xi_i\eta_0 + 15\frac{b^2}{a^2}\xi_i + 5\mu\xi_j\eta_0\right] \tag{7-31}$$

$$a_{21} = -3Hb\left[\left(2 + 3\mu + 5\frac{a^2}{b^2}\right)\xi_0\eta_j + 15\frac{a^2}{b^2}\eta_j + 5\mu\xi_0\eta_i\right] \tag{7-32}$$

$$a_{22} = Hb^2\left[2(1-\mu)\xi_0(3 + 5\eta_0) + 5\frac{a^2}{b^2}(3 + \xi_0)(3 + \eta_0)\right] \tag{7-33}$$

$$a_{23} = -15H\mu ab(\xi_i + \xi_j)(\eta_i + \eta_j) \tag{7-34}$$

$$a_{31} = 3Ha\left[\left(2 + 3\mu + 5\frac{b^2}{a^2}\right)\xi_j\eta_0 + 15\frac{b^2}{a^2}\xi_j + 5\mu\xi_i\eta_0\right] \tag{7-35}$$

$$a_{32} = -15H\mu ab(\xi_i + \xi_j)(\eta_i + \eta_j) \tag{7-36}$$

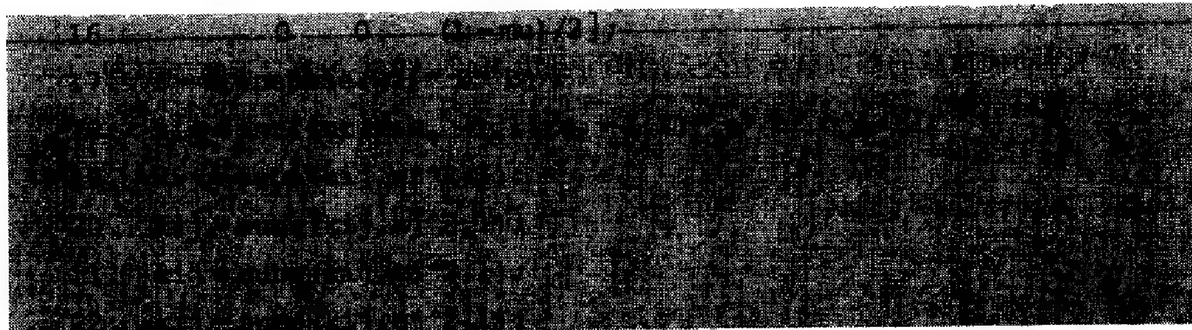
$$a_{33} = Ha^2\left[2(1-\mu)\eta_0(3 + 5\xi_0) + 5\frac{b^2}{a^2}(3 + \xi_0)(3 + \eta_0)\right] \tag{7-37}$$

式中

$$H = \frac{D}{60ab}, \quad \xi_0 = \xi_i\xi_j, \quad \eta_0 = \eta_i\eta_j \tag{7-38}$$

上面的推导过程十分繁琐，容易出错，但是如果借助 **MATLAB** 的符号运算功能，这些工作就显得不那么繁琐了，下面是用 **MATLAB** 推导矩形单元刚度矩阵的程序段。

```
1.  syms x e xi ei
2.  syms xj ej
3.  syms b a
4.  syms E mu h
5.  x0 = xi* x;
6.  e0 = ei* e;
7.  Ni = (1 + x0)* (1 + e0)* (2 + x0 + e0 - x^2 - e^2)/8;
8.  Nxi = -b* ei* (1 + x0)* (1 + e0)* (1 - e^2)/8;
9.  Nyi = a* xi* (1 + x0)* (1 + e0)* (1 - x^2)/8;
10.  N1 = [Ni Nxi Nyi];
11.  N2 = subs(subs(N1,xi,xj),ei,ej);
12.  Bi = -1/a/b* [diff(N1,x,2)* b/a; diff(N1,e,2)* a/b;
            diff(diff(N1,x,1),e,1)* 2];
13.  Bj = -1/a/b* [diff(N2,x,2)* b/a; diff(N2,e,2)* a/b;
            diff(diff(N2,x,1),e,1)* 2];
14.  D = E/(1 - mu^2)* [1  mu  0;
15.              mu  1  0
```



上面的程序段中，变量 x 和 e 是局部坐标 ξ 和 η，变量 xi, ei, xj, ej 表示 ξ_i,η_i,ξ_j 和 η_j；而 mu 则表示 Poisson 比 μ。

7.2.3 等效结点力

如果平板单元受分布横向面荷载 p 的作用，那么等效结点力是

$$f_{pi}^e = \begin{Bmatrix} f_{zi} \\ M_{\theta xi} \\ M_{\theta yi} \end{Bmatrix} = \int_{-1}^{1}\int_{-1}^{1} pN_i^T ab\,d\xi d\eta \quad (i = 1,2,3,4) \tag{7-39}$$

如果 p 在单元表面上线性分布，则可以表示成

$$p = \sum_{i=1}^{4} \overline{N}_i p_i \tag{7-40}$$

式中 $p_i(i=1,2,3,4)$ 是 4 个结点上的荷载值，N_i 是四结点矩形平面单元的形函数，即

$$N_i = (1+\xi_i\xi)(1+\eta_i\eta)/4 \quad (i=1,2,3,4) \tag{7-41}$$

那么式(7-39)的被积函数是多项式，可以积出显式如下

$$f_{pi}^e = \begin{Bmatrix} f_{zi} \\ M_{\theta xi} \\ M_{\theta yi} \end{Bmatrix} = \frac{ab}{180}\begin{Bmatrix} 45\bar{p}_1 + 7\xi_i\eta_i\bar{p}_2 + 18\xi_i\bar{P}_3 + 18\eta_i\bar{p}_4 \\ -b(15\eta_i\bar{p}_1 + \xi_i\bar{p}_2 + 5\xi_i\eta_i\bar{p}_3 + 3\bar{p}_4) \\ a(15\xi_i\bar{p}_1 + \eta_i\bar{p}_2 + 3\bar{p}_3 + 5\xi_i\eta_i\bar{p}_4) \end{Bmatrix} \quad (i=1,2,3,4) \tag{7-42}$$

式中

$$\begin{aligned} \bar{p}_1 &= p_1 + p_2 + p_3 + p_4, \quad \bar{p}_2 = p_1 - p_2 + p_3 - p_4 \\ \bar{p}_3 &= -p_1 + p_2 + p_3 - p_4, \quad \bar{p}_4 = -p_1 - p_2 + p_3 + p_4 \end{aligned} \tag{7-43}$$

当 $p=p_0$ 为常量时，$\bar{p}_1 = 4p_0$，$\bar{p}_2 = \bar{p}_3 = \bar{p}_4 = 0$，则式(7-42)可简化为

$$f_{zi} = p_0\,ab, \quad M_{\theta xi} = -\frac{p_0\,ab^2}{3}\eta_i, \quad M_{\theta yi} = \frac{p_0 a^2 b}{3}\xi_i \quad (i=1,2,3,4) \tag{7-44}$$

线性分布面荷载的等效结点力公式同样可利用 **MATLAB** 进行符号运算导得。

```
1. syms xi ei
2. syms b a
3. syms p1 p2 p3 p4
4. x0 = xi* x;
5. e0 = ei* e;
6. Ni = (1 + x0)* (1 + e0)* (2 + x0 + e0 - x^2 - e^2)/8;
7. Nxi = - b* ei* (1 + x0)* (1 + e0)* (1 - e^2)/8;
8. Nyi = a* xi* (1 + x0)* (1 + e0)* (1 - x^2)/8;
9. N = [Ni Nxi Nyi];
10. N1 = (1 - x)* (1 - e)/4 ;
11. N2 = (1 + x)* (1 - e)/4 ;
12. N3 = (1 + x)* (1 + e)/4 ;
13. N4 = (1 - x)* (1 + e)/4 ;
14. p = N1* p1 + N2* p2 + N3* p3 + N4* p4;
15. f = int( int( p* transpose(N), x, -1, 1), e, -1, 1)* a* b;
16. f = expand(f);
17. f = subs(subs(f,xi^2,1),ei^2,1) ;
18. f = simple(f)
```

如果分布面荷载具有其他更复杂的分布形式，则可以利用数值积分。

7.2.4 内力矩的计算

得到刚度矩阵和等效结点力，即可求得相应的结点位移。在板结构中，内力矩常常是人们所关心的。根据式(7-4)、式(7-5)和式(7-22)，我们得到

$$[M_x \quad M_y \quad M_{xy}]^{\mathrm{T}} = \frac{h^3}{12}\sum_{i=1}^{4} \boldsymbol{D}\boldsymbol{B}_i\boldsymbol{\delta}_i \tag{7-45}$$

式中弹性矩阵 $\boldsymbol{D}$ 取平面应变的弹性矩阵。由于矩阵 $\boldsymbol{B}_i$ 是局部坐标的函数，因此在单元内内力矩不是常量，而且由不同单元计算得到的结点内力矩一般也不一致。这与前面的平面或三维单元一样，即单元间的应力和应变是不连续的，可以通过绕结点平均法或者其他磨平方法来改善内力矩的精度。

7.3 三角形薄板单元

由于矩形单元在使用上受到平板形状的限制，因此更具灵活性的三角形

单元得到广泛重视。这里只介绍一种较为简单的三角形单元，如图 7-5 所示。

7.3.1 位移模式

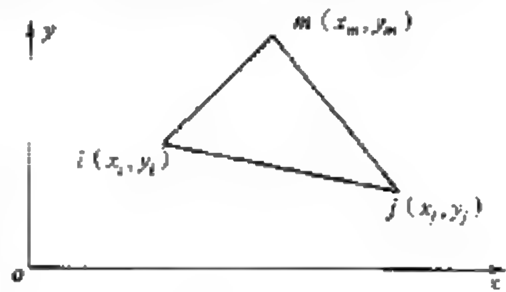


图 7-5 三角形单元

同样在三角形的 3 个顶点处，指定式(7-12)所示的结点位移列阵。由于单元形状是三角形，那么在推导单元刚度矩阵和等效结点力时，必然要在三角形上进行积分运算。根据前面平面问题三角形单元的经验知道，利用面积坐标是一个不错的选择，而且能解决整体坐标完整的三次项包含 10 项，而单元自由度只有 9 个的矛盾。

设三角形的三个结点为 i,j 和 m，相应的面积坐标记做 L_i,L_j 和 L_m。因为现在有 3 个结点，每个结点有 3 个自由度，因此位移模式中可使用 9 个参数。假设位移模式是

$$
\begin{aligned}
w = {} & \alpha_1 L_i + \alpha_2 L_j + \alpha_3 L_m + \alpha_4 L_j L_m + \alpha_5 L_m L_i + \alpha_6 L_i L_j \\
& + \alpha_7 (L_j L_m^2 - L_m L_j^2) + \alpha_8 (L_m L_i^2 - L_i L_m^2) + \alpha_9 (L_i L_j^2 - L_j L_i^2)
\end{aligned}
\tag{7-46}
$$

式中的前三项反映了刚体位移，第 3～6 项对应于常应变。为了把位移模式写成标准形式，需要得到形函数。方法是把 3 个结点的 9 个位移代入，可求解 9 个参数。但是由于结点位移中后两个是挠度 w 对整体坐标 x 和 y 的导数，而假设的位移模式是用面积坐标表示的。因此推导过程比以前稍显复杂。我们先把挠度 w 对面积坐标的导数作为结点自由度，来求它们的形函数，然后利用对面积坐标的导数与对直角坐标导数之间的关系，来确定最终我们所要的形函数[9]。

把 3 个结点的面积坐标代入式(7-46)，可立刻得到

$$
\alpha_1 = w_i, \quad \alpha_2 = w_j, \quad \alpha_3 = w_m \tag{7-47}
$$

再来计算挠度 w 对面积坐标的导数，即

$$
\begin{aligned}
\frac{\partial w}{\partial L_i} = {} & w_i - w_m - \alpha_4 L_j + \alpha_5 (L_m - L_i) + \alpha_6 L_j + \alpha_7 (L_j^2 - 2L_j L_m) \\
& + \alpha_8 (4L_i L_m - L_i^2 - L_m^2) + \alpha_9 (L_j^2 - 2L_i L_j) \\
\frac{\partial w}{\partial L_j} = {} & w_j - w_m + \alpha_4 (L_m - L_j) - \alpha_5 L_i + \alpha_6 L_i + \alpha_7 (L_m^2 + L_j^2 - 4L_j L_m) \\
& + \alpha_8 (2L_i L_m - L_i^2) + \alpha_9 (2L_i L_j - L_i^2)
\end{aligned}
\tag{7-48}
$$

这里要注意,在求导数时 L_m 不是独立变量，要把其看成 $1-L_i-L_j$ 。把 3 个结点的面积坐标代入上式，得到如下 6 个方程

$$
\begin{aligned}
&w_{,Lii}=w_i-w_m-\alpha_5-\alpha_8 && w_{,Lji}=w_j-w_m-\alpha_5+\alpha_6-\alpha_8-\alpha_9\\
&w_{,Lij}=w_i-w_m-\alpha_4+\alpha_6+\alpha_7+\alpha_9 && w_{,Ljj}=w_j-w_m-\alpha_4+\alpha_7\\
&w_{,Lim}=w_i-w_m+\alpha_5-\alpha_8 && w_{,Ljj}=w_j-w_m+\alpha_4+\alpha_7
\end{aligned}
\tag{7-49}
$$

式中,$w_{,Lij}$等表示挠度 w 对面积坐标 L_i 的导数在结点 j 的函数值。联立求解上面的 6 个方程很容易得到剩下的 6 个参数

$$
\begin{aligned}
&\alpha_4=\frac{1}{2}(w_{,Ljm}-w_{,Ljj}),\quad \alpha_5=\frac{1}{2}(w_{,Lim}-w_{,Lii})\\
&\alpha_6=\frac{1}{2}(w_{,Lij}+w_{,Lji}-w_{,Lii}-w_{,Ljj})\\
&\alpha_7=w_m-w_j+\frac{1}{2}(w_{,Ljm}+w_{,Ljj})\\
&\alpha_8=w_i-w_m-\frac{1}{2}(w_{,Lii}+w_{,Lim})\\
&\alpha_9=w_j-w_i+\frac{1}{2}(w_{,Lii}+w_{,Lij}-w_{,Lji}-w_{,Ljj})
\end{aligned}
\tag{7-50}
$$

把式(7-47)和式(7-50)代回式(7-46),得到

$$
w=[\boldsymbol{N}_i\quad \boldsymbol{N}_j\quad \boldsymbol{N}_m]\begin{Bmatrix}\bar{\boldsymbol{\delta}}_i\\ \bar{\boldsymbol{\delta}}_j\\ \bar{\boldsymbol{\delta}}_m\end{Bmatrix}
\tag{7-51}
$$

式中

$$
\begin{aligned}
&\bar{\boldsymbol{\delta}}_i=[w_i\quad w_{,Lii}\quad w_{,Lji}]^{\mathrm{T}},\quad \bar{\boldsymbol{\delta}}_j=[w_j\quad w_{,Lij}\quad w_{,Ljj}]^{\mathrm{T}}\\
&\bar{\boldsymbol{\delta}}_m=[w_m\quad w_{,Lim}\quad w_{,Ljm}]^{\mathrm{T}}
\end{aligned}
\tag{7-52}
$$

$$
\begin{aligned}
&\bar{\boldsymbol{N}}_i=[N_i\quad N_{Lii}\quad N_{Lji}],\quad \bar{\boldsymbol{N}}_j=[N_j\quad N_{Lij}\quad N_{Ljj}]\\
&\bar{\boldsymbol{N}}_m=[N_m\quad N_{Lim}\quad N_{Ljm}]
\end{aligned}
\tag{7-53}
$$

把式(7-53)的元素写出,即为

$$
\begin{aligned}
N_i &= L_i - (L_iL_j^2 - L_jL_i^2) + (L_mL_i^2 - L_iL_m^2) \\
N_{Lii} &= -\frac{1}{2}L_iL_j - \frac{1}{2}L_mL_i + \frac{1}{2}(L_iL_j^2 - L_jL_i^2) - \frac{1}{2}(L_mL_i^2 - L_iL_m^2) \\
N_{Lji} &= \frac{1}{2}L_iL_j + \frac{1}{2}(L_jL_i^2 - L_iL_j^2) \\
N_j &= L_j - (L_jL_m^2 - L_mL_j^2) + (L_iL_j^2 - L_jL_i^2) \\
N_{Lij} &= \frac{1}{2}L_iL_j + \frac{1}{2}(L_iL_j^2 - L_jL_i^2) \\
N_{Ljj} &= -\frac{1}{2}L_jL_m - \frac{1}{2}L_iL_j + \frac{1}{2}(L_jL_m^2 - L_mL_j^2) - \frac{1}{2}(L_iL_j^2 - L_jL_i^2) \\
N_m &= L_m - (L_mL_i^2 - L_iL_m^2) + (L_jL_m^2 - L_mL_j^2) \\
N_{Lim} &= \frac{1}{2}L_iL_m + \frac{1}{2}(L_iL_m^2 - L_mL_i^2) \\
N_{Ljm} &= \frac{1}{2}L_jL_m + \frac{1}{2}(L_jL_m^2 - L_mL_j^2)
\end{aligned}
\tag{7-54}
$$

下面再来把挠度对面积坐标的导数转换成对直角坐标的导数。根据复合函数的求导法则，得到

$$
\begin{aligned}
\frac{\partial w}{\partial L_i} &= \frac{\partial w}{\partial x}\frac{\partial x}{\partial L_i} + \frac{\partial w}{\partial y}\frac{\partial y}{\partial L_i} \\
\frac{\partial w}{\partial L_j} &= \frac{\partial w}{\partial x}\frac{\partial x}{\partial L_j} + \frac{\partial w}{\partial y}\frac{\partial y}{\partial L_j}
\end{aligned}
\tag{7-55}
$$

根据面积坐标与直角坐标之间的关系式（4-42）得到

$$
\frac{\partial x}{\partial L_i} = x_i - x_m = c_j, \quad \frac{\partial y}{\partial L_i} = y_i - y_m = -b_j \tag{7-56}
$$

$$
\frac{\partial x}{\partial L_j} = x_j - x_m = -c_i, \quad \frac{\partial y}{\partial L_j} = y_j - y_m = b_i \tag{7-57}
$$

把上面两式代入式(7-55)得到

$$
\begin{aligned}
\frac{\partial w}{\partial L_i} &= c_j\frac{\partial w}{\partial x} - b_j\frac{\partial w}{\partial y} = -b_j\theta_x - c_j\theta_y \\
\frac{\partial w}{\partial L_j} &= -c_i\frac{\partial w}{\partial x} + b_i\frac{\partial w}{\partial y} = b_i\theta_x + c_i\theta_y
\end{aligned}
\tag{7-58}
$$

因此，对直角坐标求导表示的结点位移与对面积坐标求导表示的结点位移之间的关系为

$$\overline{\boldsymbol{\delta}}_i = \begin{Bmatrix} w_i \\ w_{,L_{ii}} \\ w_{,L_{ji}} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -b_j & -c_j \\ 0 & b_i & c_i \end{bmatrix} \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \end{Bmatrix} = \boldsymbol{P}\boldsymbol{\delta}_i \tag{7-59}$$

式中 $\boldsymbol{\delta}_i$ 为结点位移列阵，如式(7-12)所定义，利用式(7-59)把式(7-51)改写成

$$w = [\overline{\boldsymbol{N}}_i \quad \overline{\boldsymbol{N}}_j \quad \overline{\boldsymbol{N}}_m] \begin{bmatrix} \boldsymbol{P} & 0 & 0 \\ 0 & \boldsymbol{P} & 0 \\ 0 & 0 & \boldsymbol{P} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\delta}_i \\ \boldsymbol{\delta}_j \\ \boldsymbol{\delta}_m \end{Bmatrix} = [\boldsymbol{N}_i \quad \boldsymbol{N}_j \quad \boldsymbol{N}_m] \begin{Bmatrix} \boldsymbol{\delta}_i \\ \boldsymbol{\delta}_j \\ \boldsymbol{\delta}_m \end{Bmatrix} \tag{7-60}$$

式中

$$\boldsymbol{N}_i = \overline{\boldsymbol{N}}_i \boldsymbol{P} = [N_i \quad N_{xi} \quad N_{yi}] \quad (i,j,m) \tag{7-61}$$

经过整理，可得到如下形式的形函数

$$\begin{aligned} N_i &= L_i + L_i^2 L_j + L_i^2 L_m - L_i L_j^2 - L_i L_m^2 \\ N_{xi} &= b_j L_i^2 L_m - b_m L_i^2 L_j + \frac{1}{2}(b_j - b_m) L_i L_j L_m \quad (i,j,m) \\ N_{yi} &= c_j L_i^2 L_m - c_m L_i^2 L_j + \frac{1}{2}(c_j - c_m) L_i L_j L_m \end{aligned} \tag{7-62}$$

可以验证，上列形函数及其导数具有表 7-1 所列的性质。

三角形薄板单元的形函数性质　　表 7-1

	N_i	$\frac{\partial N_i}{\partial y}$	$-\frac{\partial N_i}{\partial x}$	N_{xi}	$\frac{\partial N_{xi}}{\partial y}$	$-\frac{\partial N_{xi}}{\partial x}$	N_{yi}	$\frac{\partial N_{yi}}{\partial y}$	$-\frac{\partial N_{yi}}{\partial x}$
在结点 i	1	0	0	0	1	0	0	0	1
在结点 j、m	0	0	0	0	0	0	0	0	0

由式(7-46)的前 6 项可知，三角形薄板单元的位移模式包含刚体位移和常应变，因此是完备单元。另外，我们可以验证在相邻单元的交界线上，挠度和挠度沿交界线切向的导数是连续的，但是挠度沿交界线法向的导数却是不连续的。事实上，在任一条边上，挠度可表达成切线方向弧长 s 的三次式，并且不包含不在此边上的结点位移（因为与之对应的形函数在此边上为零）。也就是说，一条边上的挠度可以由端部两个结点处的 w 和 $\partial w/\partial s$ 所完全决定，而对于 $\partial w/\partial n$ 则不然。因此这个三角形单元是一个完备的非协调单元，更确切地称为部分协调单元。

7.3.2 刚度矩阵

在推导刚度矩阵和弯矩公式时，要计算挠度对直角坐标的导数。由于形函数是用面积坐标表示的，因此必须写出两个坐标系中偏导数的关系。仍然取 L_i、L_j 作为独立坐标，而 $L_m = 1 - L_i - L_j$ 作为 L_i 和 L_j 的函数，根据复合函数的求导数规则，并且利用坐标变换公式(4-42)，可以得到下列两个关系式

$$\begin{Bmatrix} \partial/\partial x \\ \partial/\partial y \end{Bmatrix} = \frac{1}{2\Delta}\begin{bmatrix} b_i & b_j \\ c_i & c_j \end{bmatrix}\begin{Bmatrix} \partial/\partial L_i \\ \partial/\partial L_j \end{Bmatrix} \tag{7-63}$$

和

$$\begin{Bmatrix} \partial^2/\partial x^2 \\ \partial^2/\partial y^2 \\ 2\partial^2/\partial x\partial y \end{Bmatrix} = \frac{1}{4\Delta^2}T\begin{Bmatrix} \partial^2/\partial L_i^2 \\ \partial^2/\partial L_j^2 \\ \partial^2/\partial L_i\partial L_j \end{Bmatrix} \tag{7-64}$$

式中

$$T = \begin{bmatrix} b_i^2 & b_j^2 & 2b_ib_j \\ c_i^2 & c_j^2 & 2c_ic_j \\ 2b_ic_i & 2b_jc_j & 2(b_ic_j + b_jc_i) \end{bmatrix} \tag{7-65}$$

现在把(7-60)代入几何方程(7-4)，可以得到单元应变列阵

$$\boldsymbol{\varepsilon} = \boldsymbol{B}\boldsymbol{\delta}^e = z[\boldsymbol{B}_i \quad \boldsymbol{B}_j \quad \boldsymbol{B}_m]\begin{Bmatrix} \boldsymbol{\delta}_i \\ \boldsymbol{\delta}_j \\ \boldsymbol{\delta}_m \end{Bmatrix} \tag{7-66}$$

式中

$$\boldsymbol{B}_k = -\begin{bmatrix} \boldsymbol{N}_{k'xx} \\ \boldsymbol{N}_{k'yy} \\ \boldsymbol{N}_{k'xy} \end{bmatrix} = -\frac{1}{4\Delta^2}T\begin{bmatrix} \boldsymbol{N}_{k'ii} \\ \boldsymbol{N}_{k'jj} \\ \boldsymbol{N}_{k'ij} \end{bmatrix} \quad (k = i,j,m) \tag{7-67}$$

式中符号 $\boldsymbol{N}_{k'ii}$ 等表示形函数 $\boldsymbol{N}_k$ 对面积坐标 L_i 的两次偏导数。则单元刚度矩阵的子矩阵为

$$\boldsymbol{K}_{rs}^e = \frac{h^3}{12}\iint_{\Omega} \boldsymbol{B}_r^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}_s \mathrm{d}x\mathrm{d}y \quad (r,s = i,j,m) \tag{7-68}$$

由式(7-62)得到，形函数都是面积坐标的多项式，因此应变矩阵 $\boldsymbol{B}$ 也是面积坐标的多项式。

所以式(7-68)中的积分是对面积坐标的多项式在三角形上进行积分，由

面积坐标的积分公式，能够得到显式的刚度矩阵，但是具体公式很长，可参见本章参考文献[9]。事实上，采用数值积分也很方便。为此，先把在三角形区域 Ω 上的积分化为

$$\iint_{\Omega} f(L_1, L_2, L_3)\,\mathrm{d}x\mathrm{d}y = 2\Delta \int_0^1 \int_0^{1-L_1} f(L_1, L_2, L_3)\,\mathrm{d}L_1 \mathrm{d}L_2 \tag{7-69}$$

而对于上式右边的积分可用公式[10]

$$\int_0^1 \int_0^{1-L_1} f(L_1, L_2, L_3)\,\mathrm{d}L_1 \mathrm{d}L_2 = \sum_{i=1}^{n} W_i f(L_{i1}, L_{i2}, L_{i3}) \tag{7-70}$$

式中，n 是积分点数，W_i 是权系数，(L_{i1}, L_{i2}, L_{i3}) 是积分点。为了方便，表 7-2 列出了权系数和积分点坐标。

三角形高斯积分的积分点和权系数　　表 7-2

积分点数	面积坐标	权系数
1	(1/3, 1/3, 1/3)	1/2
3	(1/2, 1/2, 0)	1/6
	(0, 1/2, 1/2)	1/6
	(1/2, 0, 1/2)	1/6
7	(1/3, 1/3, 1/3)	9/40
	(1/2, 1/2, 0)	2/30
	(0, 1/2, 1/2)	2/30
	(1/2, 0, 1/2)	2/30
	(1, 0, 0)	1/40
	(0, 1, 0)	1/40
	(0, 0, 1)	1/40

由于形函数是面积坐标的二次多项式，则应变矩阵是面积函数的一次多项式，因此式(7-68)中三角形上的被积函数是面积坐标的二次多项式，利用三点高斯积分公式可得到精确的结果。

下面给出了利用数值积分计算三角形单元刚度矩阵的程序段。

```
1.  function k = [illegible]( ie )
2.  %  计算薄板三角形单元的刚度矩阵
3.  %  输入参数:
4.  %     ie -- 单元号
5.  %  返回值:
6.  %     k  -- 单元刚度矩阵
7.      global gNode gElement
8.  [illegible]
9.      x = gNode( gElement( ie, 1:3), 1) ;
10.     y = gNode( gElement( ie, 1:3), 2) ;
```

```
11.     i = 1; j = 2; m = 3;
12.     ai = x(j)* y(m) - x(m)* y(j) ;
13.     aj = x(m)* y(i) - x(i)* y(m) ;
14.     am = x(i)* y(j) - x(j)* y(i) ;
15.     delta2 = ai + aj + am ;
16.     %
17.     k = zeros( 9, 9) ;
18.     L = [1/2  1/2  0;
19.          0    1/2  1/2;
20.          1/2  0    1/2] ;
21.     W = [1/6  1/6  1/6] ;
22.     D = MatrixD( ie) ;
23.     for i = 1:length(W)
24.             B = MatrixB( ie, L(i,:)) ;
25.             k = k + W(i)* transpose(B)* D* B* delta2 ;
26.     end
27. return
28.
29. function B = MatrixB( ie, L)
30. %   计算薄板三角形单元的应变矩阵
31. %   输入参数:
32. %      ie ---  单元号
33. %   返回值:
34. %      B  ---  单元应变矩阵
35.     global gNode gElement
36.
37.     x = gNode( gElement( ie, 1:3), 1) ;
38.     y = gNode( gElement( ie, 1:3), 2) ;
39.     i = 1; j = 2; m = 3;
40.     ai = x(j)* y(m) - x(m)* y(j) ;
41.     aj = x(m)* y(i) - x(i)* y(m) ;
42.     am = x(i)* y(j) - x(j)* y(i) ;
43.     bi = y(j) - y(m) ;
44.     bj = y(m) - y(i) ;
45.     bm = y(i) - y(j) ;
46.     ci = -(x(j) - x(m)) ;
```

```
47.        cj = - (x(m) - x(i)) ;
48.        cm = - (x(i) - x(j)) ;
49.        delta2 = ci + cj + cm ;
50.        T = [  bi^2          bj^2            2* bi* bj
51.               ci^2          cj^2            2* ci* cj
52.               2* bi* ci     2* bj* cj       2* (bi* cj + bj* ci) ] ;
53.       Li = L(1); Lj = L(2); Lm = L(3) ;
54.       Nii = [ -4* Lj - 12* Li + 6, ...
55.               -6* bj* Li + 2* bj + 3* bj* Lj + bm* Lj, ...
56.               2* cj - 6* cj* Li - 3* cj* Lj - cm* Lj, ...
57.               -4* Lj, ...
58.               (-bm + bi)* Lj, ...
59.               - (cm - ci)* Lj, ...
60.               -6 + 12* Li + 8* Lj, ...
61.               bi* Lj - [illegible] ...
62.               ci* Lj + 4* cj - 6* cj* Li - 3* cj* Lj ] ;
63.       Njj = [ -4* Li, ...
64.               - (bj - bm)* Li, ...
65.               (-cj + cm)* Li, ...
66.               6 - 4* Li - 12* Lj, ...
67.               bm* Li + 6* bi* Lj - 2* bi + 3* bi* Lj, ...
68.               cm* Li + 6* ci* Lj - 2* ci + 3* ci* Li, ...
69.               8* Li - 6 + 12* Lj, ...
70.               6* bi* Lj - 4* bi + 3* bi* Li - bj* Li, ...
71.               6* ci* Lj - 4* ci + 3* ci* Li - cj* Li ] ;
72.       Nij = [ -4* Li - 4* Lj + 2, ...
73.               -3* bj* Li [illegible] Lj - 1/2* [illegible] Lj, ...
74.               -3* cj* Li [illegible] Lj - 1/2* [illegible] Lj, ...
75.               -4* Li - 4* Lj + 2, ...
76.               bm* Lj + 3* bi* Lj [illegible] Li - 1/2* bi + bm* Li, ...
77.               cm* Lj + 3* ci* Lj + 1/2* [illegible] - cm* Li - 1/2* ci + ci* Li, ...
78.               -4 + 8* Li + 8* Lj, ...
79.               3* bi* Lj - 3/2* bi + bi* Li - 3* bj* Li + 3/2* bj - bj* Lj, ...
80.               3* ci* Lj - 3/2* ci + ci* Li - 3* cj* Li + 3/2* cj - cj* Lj ] ;
81.      B = -1/delta2^2* T* [Nii, Njj, Nij] ;
82.  return
```

```
83.
84.  function D = MatrixD( ie)
85.  %   计算薄板三角形单元的弹性矩阵
86.  %   输入参数:
87.  %       ie -- 单元号
88.  %   返回值:
89.  %       D -- 单元弹性矩阵
90.      global gElement gMaterial
91.      E = gMaterial( gElement( ie, 4), 1) ;
92.      mu = gMaterial( gElement( ie, 4), 2) ;
93.      h = gMaterial( gElement( ie, 4), 3) ;
94.      D = E* h^3/12/(1 - mu^2)* [1      mu      0
95.                                mu     1       0
96.                                0      0       (1-mu)/2] ;
97.  return
```

7.3.3 等效结点力

设单元受横向的分布荷载 p 的作用，于是等效结点力是

$$f_{pi}=\begin{Bmatrix} f_{zi} \\ M_{\theta xi} \\ M_{\theta yi} \end{Bmatrix}=\iint_{\Omega} pN_i^{\mathrm{T}}\mathrm{d}x\mathrm{d}y \quad (i,j,m) \tag{7-71}$$

如果分布荷载在单元内是线性变化的，即

$$p=L_ip_i+L_jp_j+L_mp_m \tag{7-72}$$

式中，p_i、p_j 和 p_m 分别是结点 i、j 和 m 上的分布荷载大小。把上式代入式（7-70），并利用面积坐标积分公式，得到

$$\begin{Bmatrix} f_{zi} \\ M_{\theta xi} \\ M_{\theta yi} \end{Bmatrix}=\frac{\Delta}{360}\begin{Bmatrix} 64p_i+28(p_j+p_m) \\ 7(b_j-b_m)p_i+(3b_j-5b_m)p_j+(5b_j-3b_m)p_m \\ 7(c_j-c_m)p_i+(3c_j-5c_m)p_j+(5c_j-3c_m)p_m \end{Bmatrix} \quad (i,j,m) \tag{7-73}$$

式中 Δ 是三角形单元的面积。如果分布载荷为常数，即 $p=p_0$，则上式简化为

$$\begin{Bmatrix} f_{zi} \\ M_{\theta xi} \\ M_{\theta yi} \end{Bmatrix}=\frac{\Delta p_0}{24}\begin{Bmatrix} 8 \\ b_j-b_m \\ c_j-c_m \end{Bmatrix} \quad (i,j,m) \tag{7-74}$$

内力矩的计算公式在形式上跟矩形板单元的类似，即

$$\begin{bmatrix} M_x & M_y & M_{xy} \end{bmatrix}^{\mathrm{T}} = \frac{h^3}{12}\sum_{k=i,j,m} \boldsymbol{D}\boldsymbol{B}_k\boldsymbol{\delta}_k \tag{7-75}$$

7.4 基于 Mindlin 板理论的四边形单元

前面所述的矩形单元和三角形单元都是基于 **Kirchhoff** 薄板理论的，它忽略了剪切变形的影响。由于 **Kirchhoff** 板理论要求挠度的导数连续，给构造协调单元带来了不少麻烦。为此，采用考虑剪切变形的 **Mindlin** 板理论来克服[9,11]。这种方法比较简单，精度较好，并且能利用等参变换，得到任意四边形甚至曲边四边形单元，因而实用价值较高。

7.4.1 位移模式

设有 **4～8** 结点四边形板单元，如图 **7-6** 所示。根据 **Mindlin** 板理论的假设，板内任意一点的位移由 **3** 个广义位移 w、ψ_x 和 ψ_y 完全确定。为了与有限元的结点位移相对应，采用的位移列阵为

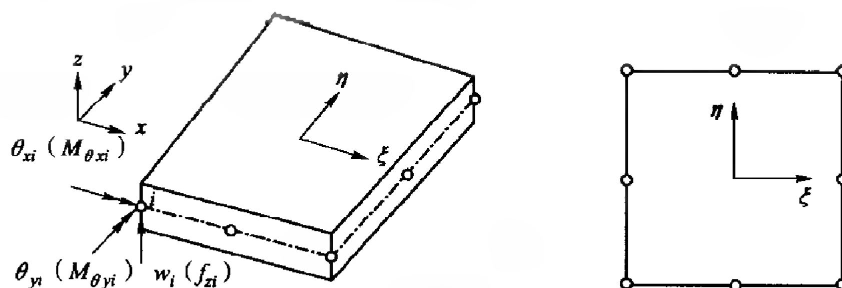


图 7-6 四边形板单元

$$\boldsymbol{u} = \begin{Bmatrix} w \\ \theta_x \\ \theta_y \end{Bmatrix} = \begin{Bmatrix} w \\ \psi_y \\ -\psi_x \end{Bmatrix} \tag{7-76}$$

式中 θ_x 和 θ_y 是跟挠度 w 无关的独立的转角位移。其结点位移列阵为

$$\boldsymbol{\delta}_i = \begin{bmatrix} w_i & \theta_{xi} & \theta_{yi} \end{bmatrix}^{\mathrm{T}} \tag{7-77}$$

引入与第 **6** 章中相同的等参变换，即

$$x = \sum_{i=1}^{n} N_i x_i, \quad y = \sum_{i=1}^{n} N_i y_i \tag{7-78}$$

式中形函数 N_i 与第 **6** 章中平面单元的形式相同。位移采用同样的插值

$$w = \sum_{i=1}^{n} N_i w_i, \quad \theta_x = \sum_{i=1}^{n} N_i \theta_{xi}, \quad \theta_y = \sum_{i=1}^{n} N_i \theta_{yi} \tag{7-79}$$

或者写成统一的形式

$$\boldsymbol{u} = \sum_{i=1}^{n} \boldsymbol{N}_i \boldsymbol{\delta}_i = \boldsymbol{N}\boldsymbol{\delta}^e \tag{7-80}$$

式中

$$\boldsymbol{N} = [\boldsymbol{N}_1 \quad \boldsymbol{N}_2 \quad \cdots \quad \boldsymbol{N}_n], \quad \boldsymbol{N}_i = N_i \boldsymbol{I}_3 \tag{7-81}$$

7.4.2 应变、应力和单元刚度矩阵

Mindlin 板理论考虑了横向剪切变形，因此应变有 5 个分量，即

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{Bmatrix} = \begin{Bmatrix} z\, \partial\theta_y/\partial x \\ -z\, \partial\theta_x/\partial y \\ z(\partial\theta_y/\partial y - \partial\theta_x/\partial x) \\ \partial w/\partial y - \theta_x \\ \partial w/\partial x + \theta_y \end{Bmatrix} \tag{7-82}$$

把式(7-79)代入上式，得到

$$\boldsymbol{\varepsilon} = \begin{bmatrix} z\boldsymbol{B}_b \\ \boldsymbol{B}_s \end{bmatrix} \boldsymbol{\delta}^e \tag{7-83}$$

式中

$$\boldsymbol{B}_b = [\boldsymbol{B}_{b1} \quad \boldsymbol{B}_{b2} \quad \cdots \quad \boldsymbol{B}_{bn}], \quad \boldsymbol{B}_s = [\boldsymbol{B}_{s1} \quad \boldsymbol{B}_{s2} \quad \cdots \quad \boldsymbol{B}_{sn}] \tag{7-84}$$

而其中的子矩阵为

$$\boldsymbol{B}_{bi} = \begin{bmatrix} 0 & 0 & \dfrac{\partial N_i}{\partial x} \\ 0 & -\dfrac{\partial N_i}{\partial y} & 0 \\ 0 & -\dfrac{\partial N_i}{\partial x} & \dfrac{\partial N_i}{\partial y} \end{bmatrix}, \quad \boldsymbol{B}_{si} = \begin{bmatrix} \dfrac{\partial N_i}{\partial y} & -N_i & 0 \\ \dfrac{\partial N_i}{\partial x} & 0 & N_i \end{bmatrix} \quad (i = 1, 2, \cdots, n) \tag{7-85}$$

而相应的应力分量也有 5 个，它们与应变之间的关系是

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \\ \tau_{yz} \\ \tau_{xz} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{D}_b & \\ & \boldsymbol{D}_s \end{bmatrix} \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{xz} \end{Bmatrix} \tag{7-86}$$

式中弹性矩阵为

$$\boldsymbol{D}_b = \frac{E}{1-\mu^2}\begin{bmatrix} 1 & \mu & 0 \\ \mu & 1 & 0 \\ 0 & 0 & (1-\mu)/2 \end{bmatrix},\quad \boldsymbol{D}_s = \begin{bmatrix} \kappa G & 0 \\ 0 & \kappa G \end{bmatrix} \tag{7-87}$$

把式(7-83)代入式(7-86)，得到应力与结点位移之间的关系是

$$\boldsymbol{\sigma} = \begin{bmatrix} \boldsymbol{D}_b & 0 \\ 0 & \boldsymbol{D}_s \end{bmatrix}\begin{bmatrix} z\boldsymbol{B}_b \\ \boldsymbol{B}_s \end{bmatrix}\boldsymbol{\delta}^e \tag{7-88}$$

则单元刚度矩阵为

$$\begin{aligned} \boldsymbol{K}^e &= \int_{-h/2}^{h/2}\iint_{\Omega}\begin{bmatrix} z\boldsymbol{B}_b^{\mathrm{T}} & \boldsymbol{B}_s^{\mathrm{T}} \end{bmatrix}\begin{bmatrix} \boldsymbol{D}_b & 0 \\ 0 & \boldsymbol{D}_s \end{bmatrix}\begin{bmatrix} z\boldsymbol{B}_b \\ \boldsymbol{B}_s \end{bmatrix}\mathrm{d}x\mathrm{d}y\mathrm{d}z \\ &= \frac{h^3}{12}\iint_{\Omega}\boldsymbol{B}_b^{\mathrm{T}}\boldsymbol{D}_b\boldsymbol{B}_b\mathrm{d}x\mathrm{d}y + h\iint_{\Omega}\boldsymbol{B}_s^{\mathrm{T}}\boldsymbol{D}_s\boldsymbol{B}_s\mathrm{d}x\mathrm{d}y \end{aligned} \tag{7-89}$$

一般情况下，上式的积分应采用第 6 章介绍的高斯积分。

7.4.3 等效结点力

设在单元上作用有分布的横向载荷 p，则在结点 i 上的等效结点力为

$$f_{zi} = \iint_{\Omega} N_i p\mathrm{d}x\mathrm{d}y = \int_{-1}^{1}\int_{-1}^{1} N_i p|\boldsymbol{J}|\mathrm{d}\xi\mathrm{d}\eta \tag{7-90}$$

由于转角位移和挠度独立插值，因此横向载荷只产生挠度方向的等效结点力，而不产生沿转角方向的弯矩结点力。

内力矩计算公式则为

$$\begin{bmatrix} M_x & M_y & M_{xy} \end{bmatrix}^{\mathrm{T}} = \frac{h^3}{12}\sum_{i=1}^{n}\boldsymbol{D}_b\boldsymbol{B}_{bi}\boldsymbol{\delta}_i \tag{7-91}$$

7.5 平面壳体单元

壳也是应用十分广泛的一种基本构件。它的中面是一个曲面，在变形时，弯曲变形和中面变形是耦合在一起的，因此与板的变形有很大的不同。当壳体的厚度跟其他尺寸相比是小量时，称为薄壳。用有限元法在分析薄壳时，可以用薄板单元组成的折板系统去代替原来的薄壳，由平面应力状态和板弯曲应力状态加以简单组合而得到薄壳的应力状态。当单元细分时，用平面单元

组成的一个折板来近似壳体的几何形状将会得到良好的结果。这就像用直梁单元来分析拱一样。通常对于任意形状的壳体，采用三角形单元比较方便。如果在壳体上容易找到同一平面上的4个点，也可以采用四边形单元。

壳体平面单元的应力状态是由平面应力和弯曲应力叠加而成的，因此在构造壳体平面单元时，只要将前面平面单元和前一节相应的单元进行简单的组合即可。前面导出的刚度矩阵可作为建立壳体平面单元刚度矩阵的基础。由于不同的板单元在不同的平面内，最后集成整体刚度矩阵时，必须经过坐标变换成整体坐标系下的单元刚度矩阵。首先我们来讨论在局部坐标系下的刚度矩阵。

7.5.1 局部坐标系中的单元刚度矩阵

下面以三角形单元为例，来说明壳体单元刚度矩阵的构成。若把局部坐标系的 x 轴和 y 轴取在单元所在的平面内，其结点位移和结点力如图 7-7 所示。

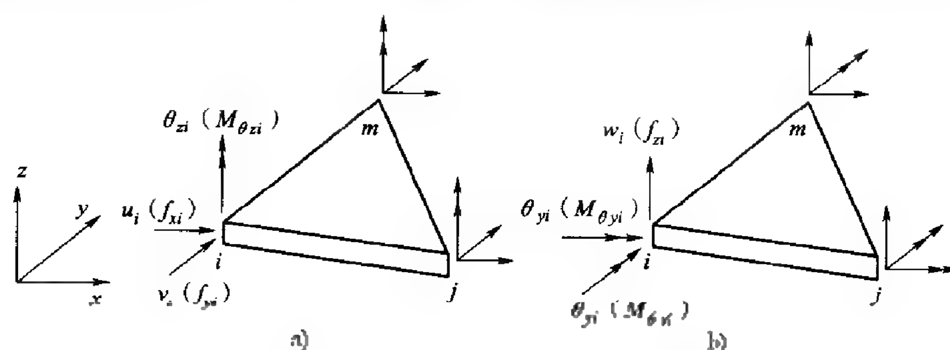


图 7-7 薄壳平面单元的结点力和结点位移

a)平面应力状态；b)弯曲应力状态

对于平面应力问题，由第 4 章可知，单元的应变状态完全取决于各结点的位移，以三角形单元为例，单元的结点力与位移之间的关系是

$$\begin{Bmatrix} \boldsymbol{f}_i^p \\ \boldsymbol{f}_j^p \\ \boldsymbol{f}_m^p \end{Bmatrix} = \boldsymbol{K}^p \begin{Bmatrix} \boldsymbol{\delta}_i^p \\ \boldsymbol{\delta}_j^p \\ \boldsymbol{\delta}_m^p \end{Bmatrix} \tag{7-92}$$

式中上标 p 表示跟平面应力相关的，$\boldsymbol{K}^p$ 是平面三角形单元的刚度矩阵，而结点力和结点位移为

$$\boldsymbol{\delta}_i^p = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix}, \quad \boldsymbol{f}_i^p = \begin{Bmatrix} f_{xi} \\ f_{yi} \end{Bmatrix} \quad (i,j,m) \tag{7-93}$$

对于弯曲应力状态，同样有结点力和结点位移的关系

$$\begin{Bmatrix} f_i^b \\ f_j^b \\ f_m^b \end{Bmatrix} = K^b \begin{Bmatrix} \delta_i^b \\ \delta_j^b \\ \delta_m^b \end{Bmatrix} \tag{7-94}$$

同样，上标 b 表示是与弯曲应力状态相关的，K^b 是三角形板单元的刚度矩阵，结点力和结点位移是

$$\delta_i^b = \begin{Bmatrix} w_i \\ \theta_{xi} \\ \theta_{yi} \end{Bmatrix}, \quad f_i^b = \begin{Bmatrix} f_{zi} \\ M_{\theta xi} \\ M_{\theta yi} \end{Bmatrix} \quad (i,j,m) \tag{7-95}$$

把平面应力与弯曲应力加以组合，单元的结点位移和结点力是

$$\delta_i = \begin{Bmatrix} u_i \\ v_i \\ w_i \\ \theta_{xi} \\ \theta_{yi} \\ \theta_{zi} \end{Bmatrix}, \quad f_i = \begin{Bmatrix} f_{xi} \\ f_{yi} \\ f_{zi} \\ M_{\theta xi} \\ M_{\theta yi} \\ M_{\theta zi} \end{Bmatrix} \quad (i,j,m) \tag{7-96}$$

虽然转角 θ_{zi} 在局部坐标系下不影响单元的应力状态，但是为了便于以后把局部坐标系的刚度矩阵转换为整体坐标系的刚度矩阵，我们特地把 θ_{zi} 也包括在结点位移中，并在结点力中相应地包括一个虚拟弯矩 $M_{\theta zi}$。

这样，薄壳单元的结点力和结点位移之间的关系可写成

$$\begin{Bmatrix} f_i \\ f_j \\ f_m \end{Bmatrix} = K^e \begin{Bmatrix} \delta_i \\ \delta_j \\ \delta_m \end{Bmatrix} \tag{7-97}$$

或简单地写成统一形式

$$f^e = K^e \delta^e \tag{7-98}$$

式中单元刚度矩阵就是在组合应力状态下的单元刚度矩阵。由于平面应力状态下的结点力与弯曲应力状态下的结点位移互不影响，弯曲应力状态的结点力与平面应力状态下的结点位移也互不影响，所以组合应力状态下的单元刚

度矩阵 $\boldsymbol{K}^e$ 的子矩阵可以写成

$$\boldsymbol{K}_{rs}=\left[\begin{array}{cc:ccc:c} \multicolumn{2}{c:}{\boldsymbol{K}^p_{rs}} & 0 & 0 & 0 & 0\\ & & 0 & 0 & 0 & 0\\ \hdashline 0 & 0 & & & & 0\\ 0 & 0 & & \boldsymbol{K}^b_{rs} & & 0\\ 0 & 0 & & & & 0\\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right]\quad (r,s=i,j,m) \tag{7-99}$$

式中的子矩阵 $\boldsymbol{K}^p_{rs}$ 和 $\boldsymbol{K}^b_{rs}$ 分别是平面应力问题和薄板弯曲问题的相应子矩阵。因此，平面应力问题和薄板弯曲问题中的三角形单元矩阵组合起来，即可得到组合应力状态下的三角形单元的刚度矩阵。对于矩形薄壳单元，这种组合也是适用的，只是矩阵的阶数是 24×24 的。

7.5.2 坐标转换

由于在单元的局部坐标系中已经得到了单元的刚度矩阵，为了建立薄壳的整体刚度矩阵，必须把不同平面内的单元刚度系数在结点处加以集合，因此要再另外确定一个统一的整体坐标系，并把各单元在局部坐标系中的刚度矩阵转换到整体坐标系中去。

现在用 x、y、z 表示整体坐标，局部坐标用 x'、y'、z' 表示。在局部坐标系中，结点 i 的位移和结点力用 $\boldsymbol{\delta}_i$ 和 $\boldsymbol{f}_i$ 表示。在整体坐标系中，用 $\boldsymbol{\delta}'_i$ 和 $\boldsymbol{f}'_i$ 表示，则它们的关系为

$$\boldsymbol{\delta}'_i=\boldsymbol{L}\boldsymbol{\delta}_i,\quad \boldsymbol{f}'_i=\boldsymbol{L}\boldsymbol{f}_i \tag{7-100}$$

式中变换矩阵为

$$\boldsymbol{L}=\begin{bmatrix}\boldsymbol{\lambda} & 0\\ 0 & \boldsymbol{\lambda}\end{bmatrix},\quad \boldsymbol{\lambda}=\begin{bmatrix} l_1 & l_2 & l_3\\ m_1 & m_2 & m_3\\ n_1 & n_2 & n_3\end{bmatrix} \tag{7-101}$$

而 l_i、m_i 和 $n_i(i=1,2,3)$ 分别是局部坐标系的 x'轴、y'轴和 z'轴在整体坐标系中的方向余弦。若把整个单元在整体坐标系中的结点位移和结点力分别用 $\boldsymbol{\delta}^e$ 和 $\boldsymbol{f}^e$ 表示，而局部坐标系下用 $\boldsymbol{\delta}'^e$ 和 $\boldsymbol{f}'^e$ 表示，则变换式为

$$\boldsymbol{\delta}'^e=\boldsymbol{T}\boldsymbol{\delta}^e,\quad \boldsymbol{f}'^e=\boldsymbol{T}\boldsymbol{f}^e \tag{7-102}$$

式中

$$\boldsymbol{T}=\begin{bmatrix}\boldsymbol{L} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{L} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{L}\end{bmatrix} \tag{7-103}$$

与梁单元类似，可以得到在整体坐标系下薄壳单元的刚度矩阵

$$K^e = T^T K'^e T \tag{7-104}$$

有一个特殊情况必须注意，如果交汇于一个结点的各单元都在一个平面内，由于式(7-99)中的第6行和第6列全部为零，在局部坐标系中，该结点的第6个平衡方程将是 0 = 0。这在理论上没有什么问题，但是在程序设计中，如不加处理，将导致刚度矩阵奇异。为了排除这一困难，可以在该结点的第6个自由度上，给以任意的刚度系数而不影响计算结果。

7.5.3　局部坐标的方向余弦

下面以三角形单元为例说明。设三角形3个结点的坐标在整体坐标系 $Oxyz$ 中表示为

$$\boldsymbol{x}_i = \begin{Bmatrix} x_i \\ y_i \\ z_i \end{Bmatrix} \quad (i,j,m) \tag{7-105}$$

为了方便，取三角形 ij 边为局部坐标系的 x' 轴，结点 i 为原点，局部坐标系的 y' 轴在三角形的平面内，而 z' 轴垂直于该平面，并使结点 ijm 符合右手螺旋法则，如图 7-8 所示。

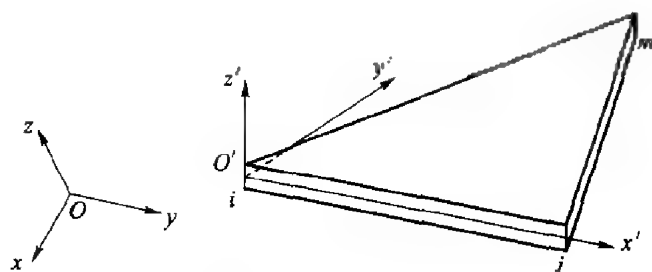


图 7-8　三角形单元的局部和整体坐标系

则 x' 方向在整体坐标系下的单位矢量是

$$\boldsymbol{e}_1 = \frac{1}{|\boldsymbol{x}_j - \boldsymbol{x}_i|}(\boldsymbol{x}_j - \boldsymbol{x}_i) \tag{7-106}$$

局部坐标系的 z' 轴垂直于三角形平面，因此沿它的矢量为

$$\boldsymbol{v}_z = (\boldsymbol{x}_j - \boldsymbol{x}_i) \times (\boldsymbol{x}_m - \boldsymbol{x}_i) \tag{7-107}$$

该矢量的长度是三角形面积的两倍，因此，z' 方向在整体坐标系下的单位矢量

是

$$e_3 = \frac{1}{2\Delta} v_z \tag{7-108}$$

那么，局部坐标系的 y' 方向在整体坐标系下的单位矢量是

$$e_2 = e_3 \times e_1 \tag{7-109}$$

则式（7-101）中的 λ 由下式确定

$$\lambda^{\mathrm{T}} = [e_1 \quad e_2 \quad e_3] \tag{7-110}$$

在局部坐标系下计算单元刚度矩阵时要使用局部坐标系下的结点坐标。根据局部坐标系 $O'x'y'z'$ 的定义，则立刻得到结点 i 和 j 的局部坐标为

$$x'_i = [0 \quad 0 \quad 0]^{\mathrm{T}}, \quad x'_j = \begin{bmatrix} 0 & |x_j - x_i| & 0 \end{bmatrix}^{\mathrm{T}} \tag{7-111}$$

而结点 m 的局部坐标为

$$x'_m = \lambda^{\mathrm{T}} (x_m - x_i) \tag{7-112}$$

其他单元局部坐标系的方向余弦可用类似方法得到。

7.6 算例

7.6.1 均布载荷作用下的矩形板

计算矩形板在均布荷载 p 作用下的挠度和弯矩，并考虑了四边简支和固支的两种边界条件。由于对称性，取板的 1/4 划分网格，采用矩形单元。首先考察单元的收敛性能。定义挠度和内力矩系数

$$\alpha = \frac{w_0 D}{pa^4}, \quad \beta = \frac{M_0}{pa^2} \tag{7-113}$$

式中 w_0 和 M_0 分别是板中心的挠度和宽度方向的内力矩，a 是板的宽度。在下面的算例中，材料的泊松比取 $\mu = 0.3$。

表 7-3 给出了方板在均布载荷作用下的挠度系数和内力矩系数随网格细分的变化，从表中可以发现，矩形板单元的结果逐渐收敛于解析解的结果[6]。表 7-4 和表 7-5 分别给出了不同长宽比（b/a）的矩形板在四边简支和四边固支条件下的挠度系数和内力系数，并与解析解的结果进行了比较。计算时的

网格采用 16×16。

方板在四边简支和四边固支条件下的挠度和内力系数(μ=0.3)　表 7-3

网格	四边简支		四边固支	
	α	β	α	β
1×1	0.0051966	0.0736191	0.0014796	0.0461648
2×2	0.0043304	0.0521323	0.0014033	0.0277831
3×3	0.0041814	0.0497384	0.0013323	0.0249581
4×4	0.0041293	0.0489183	0.0013039	0.0240477
5×5	0.0041052	0.0485448	0.0012903	0.0236320
6×6	0.0040921	0.0483428	0.0012828	0.0234084
7×7	0.0040842	0.0482214	0.0012782	0.0232742
8×8	0.0040791	0.0481427	0.0012752	0.0231873
9×9	0.0040756	0.0480888	0.0012731	0.0231279
10×10	0.0040731	0.0480503	0.0012716	0.0230855
11×11	0.0040712	0.0480218	0.0012706	0.0230541
12×12	0.0040698	0.0480002	0.0012697	0.0230303
13×13	0.0040687	0.0479833	0.0012691	0.0230117
14×14	0.0040678	0.0479700	0.0012686	0.0229970
15×15	0.0040671	0.0479592	0.0012681	0.0229852
16×16	0.0040665	0.0479504	0.0012678	0.0229754
解析解[6]	0.00406	0.0479	0.00126	0.0231

四边简支的矩形板在不同长宽比下的挠度和内力矩系数(μ=0.3)　表 7-4

b/a	挠度系数 α		内力矩系数 β	
	有限元	解析解[6]	有限元	解析解[6]
1.0	0.00407	0.00406	0.0480	0.0479
1.1	0.00487	0.00485	0.0556	0.0554
1.2	0.00566	0.00564	0.0628	0.0627
1.3	0.00640	0.00638	0.0695	0.0694
1.4	0.00709	0.00705	0.0757	0.0755
1.5	0.00773	0.00772	0.0813	0.0812
1.6	0.00832	0.00830	0.0864	0.0862
1.7	0.00885	0.00883	0.0909	0.0908
1.8	0.00933	0.00931	0.0950	0.0948
1.9	0.00976	0.00974	0.0986	0.0985
2.0	0.01014	0.01013	0.1019	0.1017

四边简支的矩形板在不同长宽比下的挠度和内力矩系数($\mu=0.3$)　表 7-5

b/a	挠度系数 α		内力矩系数 β	
	有限元	解析解[6]	有限元	解析解[6]
1.0	0.00127	0.00126	0.0230	0.0231
1.1	0.00151	0.00150	0.0268	0.0264
1.2	0.00173	0.00172	0.0301	0.0299
1.3	0.00192	0.00191	0.0328	0.0327
1.4	0.00207	0.00207	0.0351	0.0349
1.5	0.00220	0.00220	0.0369	0.0368
1.6	0.00230	0.00230	0.0383	0.0381
1.7	0.00239	0.00238	0.0394	0.0392
1.8	0.00245	0.00245	0.0402	0.0401
1.9	0.00250	0.00249	0.0408	0.0407
2.0	0.00254	0.00254	0.0413	0.0412

7.6.2 均布载荷作用下的斜板

$$\alpha_i=\frac{w_i D}{pb^4},\quad \beta_i=\frac{M_i}{pb^2}\quad (i=0,1) \tag{7-114}$$

斜桥常常可以简化成一对边简支另一对边自由的斜板，如图 7-9 所示。设荷载为均布力 p，采用三角形单元，材料的泊松比 $\mu=0.2$。图中 w_0 和 M_0 分

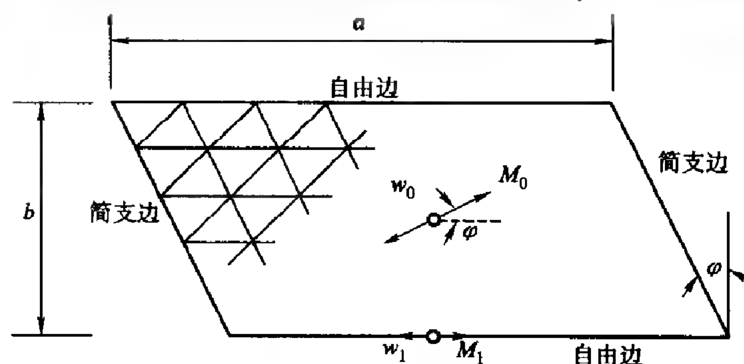


图 7-9　对边简支对边自由的斜板

别表示板中心的挠度和内力矩，内力矩方向如图所示。 w_1 和 M_1 分别表示自由边上的最大挠度和最大 x 方向内力矩，并定义如下挠度和内力矩系数。

表 7-6 列出了在不同斜角下的挠度系数和内力矩系数，并与本节参考文献[6]给出的用有限差分法的结果进行了比较。

板中心的挠度和内力矩很容易得到，但是自由边上挠度和内力矩最大值出现的位置事先并不知道。我们采用的办法是把自由边上的所有挠度和内力矩都取出来，并利用三次样条进行加密插值，最后搜索最大值。具体实现方法可参考本书所附的源程序 exam7_2.m。

斜板的挠度和内力矩系数　　表 7-6

斜角 φ	a/b	计算方法	α_0	α_1	β_0	β_1
0°	2	有限元	0.211429	0.220405	0.493691	0.505721
		有限差分[6]	0.214	0.224	0.495	0.508
30°	1.92	有限元	0.114002	0.124626	0.356945	0.334263
		有限差分[6]	0.1183	0.1302	0.368	0.367
45°	2	有限元	0.070621	0.086340	0.281261	0.230399
		有限差分[6]	0.0708	0.0869	0.291	0.296
60°	2	有限元	0.018956	0.040962	0.156744	0.114585
		有限差分[6]	0.0186	0.0396	0.166	0.152

参考文献

[1] Cook R. D., Malkus D. S., Plesha M. E., Witt R. J.. Concepts and Applications of Finite Element Analysis. John Wiley & Sons Inc., 2002.

[2] Zienkiewicz, O. C.. The Finite Element Method. 3rd ed.. London: McGraw-Hill, 1977.

[3] Hrabok M. M., Hrudey T. M.. A review and catalog of plate bending finite elements. Computers and Structures, 1984, 19(3): 479-495.

[4] Gilewski W., Radwanska M.. A survey of finite element models for the analysis of moderately thick shells. Finite Element in Analysis and Design, 1991, 9(1): 1-21.

[5] Yang H. T. Y., Saigal S., Masud A., Kapania R. K.. A survey of recent shell finite ele-

ments. International Journal for Numerical Methods in Engineering, 2000, 47(1-3): 101-127.

[6] Timoshenko S. P., Woinowsky-Krieger S.. Theory of plates and shells. 2nd ed. New York: McGraw-Hill, 1959.

[7] Mindlin R. D.. Influence of rotary inertia and shear on flexural motions of isotropic elastic plates. ASME Journal of Applied Mechanics, 1951, 18(1): 31-38.

[8] Reissner E.. On Bending of elastic plates. Quarterly of Applied Mathematics, 1947, 5(1): 55-68.

[9] 丁皓江，何福保，谢贻权，徐兴. 弹性与塑性力学中的有限单元法. 北京：机械工业出版社.

[10] Cowper G R.. Gaussian Quadrature Formulas for Triangles. International Journal of Numerical Methods in Engineering, 1973, 7 : 405-408.

[11] 朱伯芳. 有限单元法原理与应用. 2版. 北京：中国水利水电出版社

第8章 结构的振动、稳定和动力响应

结构受到动荷载作用产生振动，会影响舒适与安全，还可能引起结构局部疲劳，甚至使结构破坏，因此必须进行动力分析。最著名的例子就是美国的 **Tacoma** 桥因风激扭转颤振而倒塌。地震是造成结构破坏的另一种主要动荷载。地震中人员伤亡和财产损失的主要原因是结构的倒塌，所以加强结构动力分析，改进设计以提高结构抗振能力是防灾减灾的主要手段。

有限元对结构进行动力分析，一般分为振动特性分析和动力响应分析两种。结构的振动特性反映结构的固有特性，是研究动力问题的基础。结构的振动特性分析也称模态分析，一般包括频率和振型的计算。动力响应分析是指结构在外力作用下的强迫振动，主要求解结构的位移、速度、加速度等物理量随时间的变化情况。

弹性结构的稳定性分析随着大量高强材料和薄壁结构被采用，日益重要。世界上已发生过许多由于失稳而导致的工程事故。从力学角度来看它与前面的动力学问题毫不相干，但在数学上，它与结构的振动特性分析一样，都是一个特征值问题，因此也归入本章。

8.1 动力学方程

对结构进行动力分析时，外力和位移都是时间 t 的函数。有限元动力方程可以根据 **Hamilton** 原理建立[1]，但是根据 **d' Ale-**

mbert 原理似乎更为简洁，因为只要在外力中计及惯性力和阻尼力，便可像推导静力平衡方程一样建立动力学方程，所以大部分有限元专著均采用后者[2-4]。在动力学问题中，位移 $\boldsymbol{u}$ 是时间 t 的函数，记 $\dot{\boldsymbol{u}}$ 和 $\ddot{\boldsymbol{u}}$ 为速度和加速度向量。设单元 e 的质量密度为 ρ，则分布惯性力为 $-\rho\ddot{\boldsymbol{u}}$。假定还有与速度成正比的粘性阻尼 $-\mu\dot{\boldsymbol{u}}$，并采用如下的位移插值

$$\boldsymbol{u}=\boldsymbol{N}\boldsymbol{\delta}^e \tag{8-1}$$

则体积力的等效结点力有如下三项

$$\iiint \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{p}_v-\mu\dot{\boldsymbol{u}}-\rho\ddot{\boldsymbol{u}})\mathrm{d}V \tag{8-2}$$

第一项仍然是原来体积力的等效节点力。假设速度 $\dot{\boldsymbol{u}}$ 和加速度 $\ddot{\boldsymbol{u}}$ 也有与位移 $\boldsymbol{u}$ 一样的插值函数，即

$$\dot{\boldsymbol{u}}=\boldsymbol{N}\dot{\boldsymbol{\delta}}^e,\quad \ddot{\boldsymbol{u}}=\boldsymbol{N}\ddot{\boldsymbol{\delta}}^e \tag{8-3}$$

把上式代入式(8-2)，并记

$$\boldsymbol{C}^e=\iiint \boldsymbol{N}^{\mathrm{T}}\mu\boldsymbol{N}\mathrm{d}V \tag{8-4}$$

$$\boldsymbol{M}^e=\iiint \boldsymbol{N}^{\mathrm{T}}\rho\boldsymbol{N}\mathrm{d}V \tag{8-5}$$

式中 $\boldsymbol{C}^e$ 和 $\boldsymbol{M}^e$ 分别称为单元阻尼矩阵和质量矩阵。则式(8-2)的第2、第3项便可以写为

$$-\boldsymbol{C}^e\dot{\boldsymbol{\delta}}^e,\quad -\boldsymbol{M}^e\ddot{\boldsymbol{\delta}}^e \tag{8-6}$$

将以上等效结点力代入有限元的静力平衡方程中，便得到单元的动力方程

$$\boldsymbol{M}^e\ddot{\boldsymbol{\delta}}^e+\boldsymbol{C}^e\dot{\boldsymbol{\delta}}^e+\boldsymbol{K}^e\boldsymbol{\delta}^e=\boldsymbol{f} \tag{8-7}$$

按有限元的集合方法，最终得到整体动力方程

$$\boldsymbol{M}\ddot{\boldsymbol{\delta}}+\boldsymbol{C}\dot{\boldsymbol{\delta}}+\boldsymbol{K}\boldsymbol{\delta}=\boldsymbol{f} \tag{8-8}$$

式中 $\boldsymbol{C}$ 和 $\boldsymbol{M}$ 分别称为结构的阻尼矩阵和质量矩阵。这里要强调的是，上式是一个关于时间的常微分方程组，而原来的静力问题得到的是一个代数方程组。在这一方程中，若 $\boldsymbol{C}=0,\boldsymbol{f}=0$，则为无阻尼自由振动方程，解之可得到结构的无阻尼自由振动的频率和振型，也就是所谓的特征值问题。它与结构的线性稳定分析在数学上是一致的，若荷载 $\boldsymbol{f}$ 是时间 t 的周期函数，比如它是时间的正弦或余弦函数，此时一般只需计算稳态响应。如果 $\boldsymbol{f}$ 是时间的任意函数，则应考虑初始条件，进行瞬态响应分析，一般称为动力响应分析。

由于影响结构阻尼的因素极为复杂，缺乏式(8-4)中粘性阻尼系数 μ 的有关资料，一般来说用式(8-4)计算单元阻尼矩阵是不可行的。目前常

常假设

$$C=\alpha M+\beta K \tag{8-9}$$

α 和 β 为两个系数，由试验确定。式(8-9)所示的阻尼称为 Rayleigh 阻尼。

矩阵 $\boldsymbol{M}$ 为总体质量矩阵，$\boldsymbol{M}^e$ 为单元质量矩阵。单元质量矩阵可按式（8-5）计算，总体质量矩阵由单元质量矩阵集合而成。单元质量矩阵也可以用某种简单的等价方法把单元质量分配到各个结点上，形成一个对角质量矩阵，称为集中质量矩阵。用式(8-5)算得的称为一致质量矩阵。在很多商用有限元软件中一般都允许用户选择使用哪一种质量矩阵。显然一致质量矩阵 $\boldsymbol{M}$ 与刚度矩阵 $\boldsymbol{K}$ 一样，是稀疏的，其元素的分布情况与刚度矩阵 $\boldsymbol{K}$ 完全相同。一致质量矩阵 $\boldsymbol{M}$ 是对称正定的。这是因为对于任意非零速度向量 $\dot{\boldsymbol{\delta}}$，结构动能总是大于零，即对于任意 $\dot{\boldsymbol{\delta}}\neq 0$，有

$$\dot{\boldsymbol{\delta}}^{T}\boldsymbol{M}\dot{\boldsymbol{\delta}}>0 \tag{8-10}$$

集中质量矩阵是对角阵，但不一定是正定的。在弯曲问题中，考虑转动自由度而不考虑转动惯量时，相应的对角元为零。集中质量矩阵计算比较简单，所需的存储量也少。而且使用集中质量矩阵会使固有频率的计算值降低，使用协调单元会使结构过刚，从而使频率的计算值偏高。这两种相反的因素相抵的结果，有时会得到较好的计算值。

一般来说，采用一致质量矩阵可以得到更加精确的振型，如果单元是协调的，那么还可以有这样的结论：频率的计算值是结构真实频率的上界。

8.2 质量矩阵

从上节的讨论可知，质量矩阵是把惯性力向结点静力等效而得到的。采用不同的等效方法就会得到不同形式的质量矩阵。首先介绍平面梁单元和空间梁单元的一致质量矩阵，其次推导三角形常应变单元的一致质量矩阵，并写出相应的集中质量矩阵以进行比较。

8.2.1 平面梁单元的质量矩阵

设平面梁单元的长度为 l，密度为 ρ，截面积为 A。根据公式(8-5)，计算一致质量矩阵要用到单元的形函数，因此要先确定它。平面梁单元的形函数在第 3 章中已经得到，即公式(3-27)，具体如下

$$\boldsymbol{N}=\begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 \\ 0 & N_3 & N_4 & 0 & N_5 & N_6 \end{bmatrix} \tag{8-11}$$

式中

$$N_1 = 1 - x/l, \quad N_2 = x/l, \quad N_3 = 1 - \frac{3x^2}{l^2} + \frac{2x^3}{l^3} \tag{8-12a}$$

$$N_4 = x - \frac{2x^2}{l} + \frac{x^3}{l^2}, \quad N_5 = \frac{3x^2}{l^2} - \frac{2x^3}{l^3}, \quad N_6 = -\frac{x^2}{l} + \frac{x^3}{l^2} \tag{8-12b}$$

代入公式(8-5)后，得

$$\boldsymbol{M}^e = \rho A \int_0^l \boldsymbol{N}^{\mathrm{T}} \boldsymbol{N} \mathrm{d}x = \frac{\rho A l}{420} \begin{bmatrix} 140 & & & & & \\ 0 & 156 & & \text{对} & \text{称} & \\ 0 & 22l & 4l^2 & & & \\ 70 & 0 & 0 & 140 & & \\ 0 & 54 & 13l & 0 & 156 & \\ 0 & -13l & -3l^2 & 0 & -22l & 4l^2 \end{bmatrix} \tag{8-13}$$

上式就是平面梁单元在局部坐标系下的单元质量矩阵。同样，它也必须经过坐标变换，才能集成到整体刚度矩阵中去。坐标变换的公式与刚度矩阵的完全一样，可参考第3章。

8.2.2 空间梁单元的质量矩阵

空间梁单元还要考虑它的扭转，因此除了长度 l，密度为 ρ 和截面积为 A 外，还需要知道它的极惯性矩 J_k，来考虑梁扭转的惯性力。空间梁单元的形函数也同样在第3章中已经得到，即

$$\boldsymbol{N} = \begin{bmatrix} N_1 & 0 & 0 & 0 & 0 & 0 & N_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_3 & 0 & 0 & 0 & N_4 & 0 & N_5 & 0 & 0 & 0 & N_6 \\ 0 & 0 & N_3 & 0 & N_4 & 0 & 0 & 0 & N_5 & 0 & N_6 & 0 \\ 0 & 0 & 0 & N_1 & 0 & 0 & 0 & 0 & 0 & N_2 & 0 & 0 \end{bmatrix} \tag{8-14}$$

式中 $N_i (i = 1, 2, \cdots, 6)$ 的定义见公式(8-12)。但是与平面梁单元的情况稍有区别，不能把上式直接代入公式(8-5)进行积分运算。原因是公式(8-5)的推导过程中，假定惯性力为 $\rho \ddot{\boldsymbol{u}}$。但是空间梁单元的位移列阵是

$$\boldsymbol{u} = [u \quad v \quad w \quad \theta_x]^{\mathrm{T}} \tag{8-15}$$

它的最后一个分量是绕 x 轴的转角。为了能用式(8-5)，我们要把形函数(8-14)修改成

$$
\boldsymbol{N}=\begin{bmatrix} N_1 & 0 & 0 & 0 & 0 & 0 & N_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & N_3 & 0 & 0 & 0 & N_4 & 0 & N_5 & 0 & 0 & 0 & N_6 \\ 0 & 0 & N_3 & 0 & N_4 & 0 & 0 & 0 & N_5 & 0 & N_6 & 0 \\ 0 & 0 & 0 & rN_1 & 0 & 0 & 0 & 0 & 0 & rN_2 & 0 & 0 \end{bmatrix} \tag{8-16}
$$

式中 r 是截面上某一点到 x 轴截面中心的距离，把上式代入计算一致质量阵的公式以后,完成积分,得到空间梁单元的一致质量矩阵

$$
\boldsymbol{M}^e=\frac{\rho Al}{420}\begin{bmatrix}
140 & 0 & 0 & 0 & 0 & 0 & 70 & 0 & 0 & 0 & 0 & 0 \\
0 & 156 & 0 & 0 & 0 & 22l & 0 & 54 & 0 & 0 & 0 & -13l \\
0 & 0 & 156 & 0 & 22l & 0 & 0 & 0 & 54 & 0 & -13l & 0 \\
0 & 0 & 0 & \frac{140J_k}{A} & 0 & 0 & 0 & 0 & 0 & \frac{70J_k}{A} & 0 & 0 \\
0 & 0 & 22l & 0 & 4l^2 & 0 & 0 & 0 & 13l & 0 & -3l^2 & 0 \\
0 & 22l & 0 & 0 & 0 & 4l^2 & 0 & 13l & 0 & 0 & 0 & -3l^2 \\
70 & 0 & 0 & 0 & 0 & 0 & 140 & 0 & 0 & 0 & 0 & 0 \\
0 & 54 & 0 & 0 & 0 & 13l & 0 & 156 & 0 & 0 & 0 & -22l \\
0 & 0 & 54 & 0 & 13l & 0 & 0 & 0 & 156 & 0 & -22l & 0 \\
0 & 0 & 0 & \frac{70J_k}{A} & 0 & 0 & 0 & 0 & 0 & \frac{140J_k}{A} & 0 & 0 \\
0 & 0 & -13l & 0 & -3l^2 & 0 & 0 & 0 & -22l & 0 & 4l^2 & 0 \\
0 & -13l & 0 & 0 & 0 & -3l^2 & 0 & -22l & 0 & 0 & 0 & 4l^2
\end{bmatrix} \tag{8-17}
$$

式中 $J_k=\iint r^2\,\mathrm{d}A$ 为梁横截面的极惯性矩。

8.2.3 平面三角形常应变单元的质量矩阵

三角形单元的厚度设为 h,密度为 ρ,面积为 Δ。其形函数为

$$
\boldsymbol{N}=\begin{bmatrix} N_i & 0 & N_j & 0 & N_m & 0 \\ 0 & N_i & 0 & N_j & 0 & N_m \end{bmatrix} \tag{8-18}
$$

式中 N_i 见第 4 章的式(4-14)。设在单元内 ρ 和 h 是常数,则有

$$
\boldsymbol{M}^e=\rho h\iint \boldsymbol{N}^{\mathrm{T}}\boldsymbol{N}\mathrm{d}x\mathrm{d}y \tag{8-19}
$$

把式(8-18)代入上式,得到

$$
\boldsymbol{M}^e = \rho h \iint \begin{bmatrix}
N_i^2 & 0 & N_i N_j & 0 & N_i N_m & 0 \\
0 & N_i^2 & 0 & N_i N_j & 0 & N_i N_m \\
N_j N_i & 0 & N_j^2 & 0 & N_j N_m & 0 \\
0 & N_j N_i & 0 & N_j^2 & 0 & N_j N_m \\
N_m N_i & 0 & N_m N_j & 0 & N_m^2 & 0 \\
0 & N_m N_i & 0 & N_m N_j & 0 & N_m^2
\end{bmatrix} \mathrm{d}x\mathrm{d}y \tag{8-20}
$$

利用三角形面积坐标的积分公式，则有

$$
\iint N_r N_s \,\mathrm{d}x\mathrm{d}y = \frac{1!1!0!}{(1+1+0+2)!}2\Delta = \frac{1}{12}\Delta \quad (r,s = i,j,m) \tag{8-21}
$$

$$
\iint N_r^2 \mathrm{d}x\mathrm{d}y = \frac{2!0!0!}{(2+0+0+2)!}2\Delta = \frac{1}{6}\Delta \quad (r = i,j,m) \tag{8-22}
$$

于是，平面问题三角形单元的一致质量矩阵为

$$
\boldsymbol{M}^e = \frac{\rho h \Delta}{3} \begin{bmatrix}
1/2 & 0 & 1/4 & 0 & 1/4 & 0 \\
0 & 1/2 & 0 & 1/4 & 0 & 1/4 \\
1/4 & 0 & 1/2 & 0 & 1/4 & 0 \\
0 & 1/4 & 0 & 1/2 & 0 & 1/4 \\
1/4 & 0 & 1/4 & 0 & 1/2 & 0 \\
0 & 1/4 & 0 & 1/4 & 0 & 1/2
\end{bmatrix} \tag{8-23}
$$

将单元质量平均分配于单元的 3 个结点上便得到集中质量矩阵

$$
\boldsymbol{M}^e = \frac{\rho h \Delta}{3} \boldsymbol{I}_6 \tag{8-24}
$$

从前面几个例子可以看出质量矩阵的计算过程完全是类似的，因此有利于程序实现。对于较复杂的单元，跟单元刚度矩阵一样，一般可采用数值积分。

8.3 结构的自由振动

8.3.1 特征值与特征向量

在动力学方程(8-8)中，若令 $\boldsymbol{C}=0, \boldsymbol{f}=0$，便得到结构的无阻尼自由振动方程

$$
\boldsymbol{M}\ddot{\boldsymbol{\delta}} + \boldsymbol{K}\boldsymbol{\delta} = 0 \tag{8-25}
$$

这是一个二阶常系数齐次常微分方程，其解的形式为

$$\delta(t) = \varphi \sin \omega t \tag{8-26}$$

式中 ω 是结构自由振动的频率，把上式代入式(8-25)，便得到

$$(K - \omega^2 M)\varphi = 0 \tag{8-27}$$

这是一个 n 阶线性齐次方程组，若要有非零解，则其系数矩阵的行列式必须为零，即

$$\det(K - \omega^2 M) = 0 \tag{8-28}$$

它是 ω^2 的 n 次实系数方程，称为常微分方程组(8-25)的特征方程，它的根称为特征值，是结构频率的平方。对于实际结构，可由式(8-28)求得 n 个振动频率

$$\omega_1 \leqslant \omega_2 \leqslant \cdots \leqslant \omega_n \tag{8-29}$$

对于每个自振频率，可由式(8-27)确定一组各结点的振幅值 $\varphi_i(i = 1, 2, \cdots, n)$，它们的每一个分量之间的比值保持不变，数值大小可取任意值，称为特征向量，在振动理论中常称为结构自由振动的振型。

形如式(8-27)的特征值问题称为广义特征值问题，而形如

$$(A - \lambda I)x = 0 \tag{8-30}$$

的特征值问题称为标准特征值问题，广义特征值问题可化为标准特征值问题[4]。

特征值和特征向量有两个重要性质，非常有利于建立特征值的求解方法。

(1) 若质量矩阵正定，则有 n 个特征值，且可排序为

$$0 \leqslant \omega_1^2 \leqslant \omega_2^2 \leqslant \cdots \leqslant \omega_n^2 \tag{8-31}$$

ω_i 是结构的第 i 阶固有频率。如果再有 K 正定，则 $\omega_1 > 0$；如果 K 奇异，则 $\omega_1 = 0$。亦即结构如果没有刚体位移自由度，则所有的固有频率都大于零；如果刚体有位移自由度，则有零固有频率。

若矩阵 M 奇异，设结构在 r 个自由度上没有质量，则有 $m = n - r$ 个特征值

$$0 \leqslant \omega_1^2 \leqslant \omega_2^2 \leqslant \cdots \leqslant \omega_m^2 \tag{8-32}$$

(2)特征向量彼此正交。为了一般起见，假定矩阵 M 是奇异的，结构在 r 个自由度上没有质量，如前述结构有 $m = n - r$ 个特征值。与 m 个特征值相对应有 m 个特征向量，记为 $\varphi_1, \varphi_2, \cdots, \varphi_m$。如果 φ_i 是特征向量，$\alpha\varphi_i(\alpha \neq 0)$ 也是特征向量，即特征向量的每个分量之间的比值是固定的，但是具体数值可任意，为了消除这一不定因素，可以令

$$\varphi_i^T M \varphi_i = 1 \tag{8-33}$$

根据特征值和特征向量的定义，可以立刻得到

$$\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\varphi}_i = \omega_i^2\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\varphi}_i = \omega_i^2 \tag{8-34}$$

对于两个不同的特征根 ω_i^2 和 ω_j^2 及其对应的特征向量 $\boldsymbol{\varphi}_i$ 和 $\boldsymbol{\varphi}_j$，则有

$$\begin{aligned} \boldsymbol{K}\boldsymbol{\varphi}_i - \omega_i^2\boldsymbol{M}\boldsymbol{\varphi}_i = \boldsymbol{0} \\ \boldsymbol{K}\boldsymbol{\varphi}_j - \omega_j^2\boldsymbol{M}\boldsymbol{\varphi}_j = \boldsymbol{0} \end{aligned} \tag{8-35}$$

在上面两式中分别左乘 $\boldsymbol{\varphi}_j^{\mathrm{T}}$ 和 $\boldsymbol{\varphi}_i^{\mathrm{T}}$，对后一式取转置后，两式相减，并利用矩阵 $\boldsymbol{K}$ 和 $\boldsymbol{M}$ 的对称性，得到

$$(\omega_i^2 - \omega_j^2)\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\varphi}_j = 0 \tag{8-36}$$

由于 ω_i^2 和 ω_j^2 是两个不同的特征根，由此可得

$$\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\varphi}_j = 0 \qquad \boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\varphi}_j = 0 \tag{8-37}$$

重特征值对应的特征向量不是惟一的，s 重的特征值对应有 s 个线性无关的特征向量，它们的线性组合仍旧是该特征值所对应的特征向量。由于可以选出一组正交的单位向量作为确定的特征向量组。于是所有特征向量彼此正交，即

$$\begin{aligned} \boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\varphi}_j = \delta_{ij} \\ \boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\varphi}_j = \omega_i^2\delta_{ij} \end{aligned} \tag{8-38}$$

这些特征向量就构成一个 m 维空间。

结构的 r 个没有质量的自由度，为纯静态自由度。若存在非零位移 $\boldsymbol{\varphi}_\infty$，使

$$\boldsymbol{M}\boldsymbol{\varphi}_\infty = \boldsymbol{0} \tag{8-39}$$

这样的位移称为纯静态位移。显然有

$$\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\varphi}_\infty = 0 \tag{8-40}$$

这表示所有的特征向量与纯静态位移正交。所以由纯静态位移构成的空间与特征向量空间正交。纯静态位移空间是 r 维空间，可以找出关于 $\boldsymbol{K}$ 正交的单位向量 $\boldsymbol{\varphi}_\infty^1, \boldsymbol{\varphi}_\infty^2, \cdots, \boldsymbol{\varphi}_\infty^r$，即有

$$\begin{aligned} \boldsymbol{\varphi}_\infty^{i\mathrm{T}}\boldsymbol{K}\boldsymbol{\varphi}_\infty^j = \delta_{ij} \\ \boldsymbol{\varphi}_\infty^{i\mathrm{T}}\boldsymbol{M}\boldsymbol{\varphi}_\infty^j = 0 \end{aligned} \tag{8-41}$$

这样得到的 m 个特征向量和纯静态位移空间的 r 个基向量构成 n 维空间的一个完备的正交向量基，空间内任一向量 $\boldsymbol{\delta}$ 均可由这组基向量线性表示

$$\boldsymbol{\delta} = \sum_{i=1}^{m}\alpha_i\boldsymbol{\varphi}_i + \sum_{j=1}^{r}\beta_j\boldsymbol{\varphi}_\infty^j \tag{8-42}$$

8.3.2 矩阵特征值算法

关于矩阵特征值算法有很多，最常用的是逆迭代法和子空间迭代法。前者算法简单，比较适合于只要求得系统的很少数目特征对的情况。后者实质是将前者推广应用于若干个向量进行迭代的情况，可以用于要求得到系统多一些特征解的情况。另外，近年来，Ritz 向量直接迭代法和 Lanczos 向量的直接迭代法，由于具有更高的计算效率，引起了有限元工作者广泛的兴趣。本书只介绍逆迭代法和子空间迭代法，有兴趣的读者可参考本章参考文献[5-8]。

1. 逆迭代法

逆迭代法在计算特征值和特征向量时非常有效，它也是很多算法的基础，因此有必要先进行介绍。先假定刚度矩阵 $\boldsymbol{K}$ 是对称的，而质量矩阵可以有为零的对角元。设有初始向量 $\boldsymbol{x}_1$，及其第 $s=1,2,\cdots$步时的迭代公式为[5]

$$\boldsymbol{K}\boldsymbol{x}_{s+1}=\boldsymbol{M}\boldsymbol{x}_s \tag{8-43}$$

$$\boldsymbol{x}_{s+1}=\frac{\bar{\boldsymbol{x}}_{s+1}}{(\bar{\boldsymbol{x}}_{s+1}^{\mathrm{T}}\boldsymbol{M}\bar{\boldsymbol{x}}_{s+1})^{1/2}} \tag{8-44}$$

只要初始向量 $\bar{\boldsymbol{x}}_1$ 不关于矩阵 $\boldsymbol{M}$ 与第一个特征向量 $\boldsymbol{\varphi}_1$ 正交，也就是说 $\boldsymbol{x}_1^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\varphi}_1\neq 0$，则有当 $k\to\infty$ 时，则有 $\boldsymbol{x}_{s+1}\to\boldsymbol{\varphi}_1$。

式(8-44)是为了保证在迭代过程中，向量 $\boldsymbol{x}_{s+1}$ 的长度不要随迭代而不断地增大，导致无法收敛，同时也保证了最后得到的特征向量满足式（8-38），即

$$\boldsymbol{x}_{s+1}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{x}_{s+1}=1 \tag{8-45}$$

只要把式(8-44)代入上式即可验证。式（8-43）和(8-44)是基本的迭代公式，但是在实际计算中，还可以组织更加有效的算法如下：

(1)选取初始向量 $\boldsymbol{x}_1$，计算 $\boldsymbol{y}_1=\boldsymbol{M}\boldsymbol{x}_1$。

(2)解方程

$$\boldsymbol{K}\boldsymbol{x}_{s+1}=\boldsymbol{y}_1 \tag{8-46}$$

得到 $\boldsymbol{x}_{s+1}$

(3)计算

$$\boldsymbol{y}_{s+1}=\boldsymbol{M}\boldsymbol{x}_{s+1} \tag{8-47}$$

(4)计算近似特征值

$$\rho_{s+1}=\frac{\bar{\boldsymbol{x}}_{s+1}^{\mathrm{T}}\boldsymbol{y}_s}{\bar{\boldsymbol{x}}_{s+1}^{\mathrm{T}}\bar{\boldsymbol{y}}_{s+1}} \tag{8-48}$$

若精度满足要求

$$\frac{\rho_{s+1}-\rho_s}{\rho_{s+1}}\leqslant\varepsilon \tag{8-49}$$

则转向第 5 步计算特征向量。否则计算下次迭代的初始向量

$$y_{s+1}=\frac{\bar{y}_{s+1}}{(\bar{x}_{s+1}^{T}\bar{y}_{s+1})^{1/2}} \tag{8-50}$$

然后转向第 2 步作下一次迭代。

(5)计算特征向量

$$\varphi_1\approx\frac{\bar{x}_{s+1}}{(\bar{x}_{s+1}^{T}\bar{y}_{s+1})^{1/2}} \tag{8-51}$$

取特征值为 $\omega_1^2\approx\rho_{s+1}$,结束迭代。由此可见,逆迭代法可以得到最小的特征值及其特征向量。如果迭代方向反过来，即

$$Kx_{s+1}=Mx_s \tag{8-52}$$

则可求得最大特征值。此时要求质量阵 M 正定。这种迭代法也称为幂法或正迭代法。

逆迭代法除了求最小特征值和特征向量外，还可以扩大应用范围。和 Gram－Schmidt 正交化过程相结合，可以用来求取最低的几阶特征对。比如如果选取的初始迭代向量与前 $j-1$ 阶特征向量正交，则通过逆迭代将能得到第 j 阶的特征值和特征向量。当求得前 $j-1$ 阶特征向量以后，要做到这一点是很容易的，只要选择初始向量与前 $j-1$ 阶特征向量正交。为此，构造一个新的初向量

$$\bar{x}_1=x_1-\sum_{i=1}^{j-1}\beta_i\varphi_i \tag{8-53}$$

式中

$$\beta_i=\varphi_i^{T}Mx_1\qquad(i=1,2,\cdots,j-1) \tag{8-54}$$

这称为 Gram－Schmidt 正交化过程。

从迭代公式看，似乎这种正交化过程是一劳永逸的，一旦初始向量与前 $j-1$阶特征向量正交，则所有的迭代向量都与前 $j-1$ 阶特征向量正交。但实际计算并非如此，由于误差的原因，不可避免会产生低阶特征向量，迭代到最后还是得到最低阶的特征向量。为了克服这一困难，必须每迭代一次都进行正交化处理,不断把前 $j-1$ 阶特征向量从迭代向量中清除掉。

对于重特征值，用带正交化过程的迭代，可得出重特征值对应的一组正交特征向量。当刚度矩阵奇异时，逆迭代法中的式(8-46)无法求解,此时可利用移频法。即对于广义特征值问题(8-27),作如下变化

$$(K+\mu M)\varphi=(\omega^2+\mu)M\varphi \tag{8-55}$$

得到新的特征值问题

$$\bar{K}\varphi=\lambda M\varphi \tag{8-56}$$

式中

$$\bar{K}=K+\mu M \qquad \lambda=\omega^2+\mu \tag{8-57}$$

新特征值问题与原特征值问题的各阶特征向量相同，各阶特征值移动一个距离 μ。由于 μ 是任意的，因此总可以找到一个使得新刚度矩阵 $\bar{K}$ 为非奇异，这样，逆迭代法就可顺利进行。实际上，利用移频法不仅可以解决刚度矩阵奇异问题，而且当选取合适的移频量时，能改善逆迭代法的收敛速度和精度[4]。

最后还要提一下初始向量怎么选取的问题。初始向量选得好与坏，对迭代的效率影响极大。显然，要求初始向量选得尽可能接近所求的振型。一个简单的选取方法是，把结构简单地考虑为各个自由度是互不耦合的，认为

$$K=\mathrm{diag}[k_{11} \quad k_{22} \quad \cdots \quad k_{nn}] \tag{8-58}$$

$$M=\mathrm{diag}[m_{11} \quad m_{22} \quad \cdots \quad m_{nn}] \tag{8-59}$$

特征方程便为非耦合形式

$$k_{ii}\delta_i=\lambda_i m_{ii}\delta_i \tag{8-60}$$

所以 $\lambda_i=k_{ii}/m_{ii},\delta_i=1$。将 λ_i 从小到大排序，得各阶近似特征向量。设 λ_i 排在第 j 位，记 $\lambda_i^j=k_{ii}/m_{ii},\delta_i^j=1$。

一般取第一阶初始向量为

$$Mx_1^0=[m_{11} \quad m_{22} \quad \cdots \quad m_{nn}]^{\mathrm{T}} \tag{8-61}$$

取第 $j+1$ 阶初始向量为

$$Mx_{j+1}^0=\begin{bmatrix} \overset{1}{0} & \overset{2}{0} & \overset{\cdots}{\cdots} & 0 & \overset{i}{1} & 0 & \overset{\cdots}{\cdots} & \overset{n}{0} \end{bmatrix}^{\mathrm{T}} \quad (j=1,2,\cdots) \tag{8-62}$$

下面给出逆迭代法的 MATLAB 程序，可以发现程序的实现十分简单。

```
1.  function [V,D] = Inv... (K, M, epsilon) [illegible]
2.  [illegible]
3.  [illegible]
4.  %  [illegible]
5.  [illegible] 输入参数 [illegible]
6.  [illegible]
7.  %   epsilon —— 计算要求的精度
```

```
14.    while 1
15.        s = s+1;
16.        xs1 = K\ys;
17.        ys1 = M*xs1;
18.        if s == 1
19.          ro = transpose(xs1)*ys/(transpose(xs1)*ys1);
20.        else
21.          ro1 = transpose(xs1)*ys/(transpose(xs1)*ys1);
22.          if abs(ro1-ro)/ro1 <= epsilon
23.            break;
24.          else
25.            ro = ro1;
26.          end
27.        end
28.        ys = ys1/(transpose(xs1)*ys1)^(1/2);
29.    end
30.    e = ro1;
31.    v = xs1/(transpose(xs1)*ys1)^(1/2);
```

2. 子空间迭代法

子空间迭代法是求解大型矩阵特征值问题的最常用且有效的方法之一，它适合于求解部分特征解，被广泛应用于结构动力学的有限元分析中。实际上，可以把子空间迭代法看作是逆迭代法的推广，即同时进行多个特征对的迭代。

对于广义特征值问题(8-27)，若有一组 q 个线性无关的向量 $\boldsymbol{X} = [\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \cdots \quad \boldsymbol{x}_q]$，则可构造相应的特征值问题

$$\boldsymbol{K}a = \rho \boldsymbol{M}a \tag{8-63}$$

式中

$$\overline{K} = X^{T} KX \tag{8-64}$$

$$\overline{M} = X^{T} MX \tag{8-65}$$

解特征值问题(8-63)，得特征对 ρ_i 及 $\psi_i(i=1,2,\cdots,q)$。由这 q 个特征对可求得特征值问题(8-27)的近似解：$\lambda_i \approx \rho_i, \varphi_i \approx X\psi_i(i=1,2,\cdots,q)$。事实上，如果向量基 X 所构成的空间刚好是特征值问题式(8-27)的 q 阶特征向量所构成的空间，则所得到的便是精确解。但是在实际运算中，这一点是做不到的，因此只能求得近似解。

子空间迭代法的流程如下：

选定初始向量基 $X_1 = [x_1^{(1)} \quad x_2^{(1)} \quad \cdots \quad x_q^{(1)}]$，并计算 $Y_1 = MX_1$



对于 $s=1,2,\cdots$ 进行循环
求解 $KX_{s+1} = Y_1$
计算 $\overline{K} = X_{s+1}^{T} Y_S$
计算 $Y_2 = MX_{s+1}$
计算 $\overline{M} = X_{s+1}^{T} Y_2$
求解广义特征问题 $\overline{K}\psi = \rho\overline{M}\psi$ 的全部特征值和特征向量
如果特征值的精度满足要求，则终止循环
计算 $Y_1 = Y_2\psi$，这里 $\psi = [\psi_1 \quad \psi_2 \quad \cdots \psi_q]$



计算原特征值问题的特征值 λ_i 和特征向量 φ_i 为 $\lambda_i \approx \rho_i, \varphi_i \approx X_{s+1}\psi_i(i=1,2,\cdots,q)$

通过以上分析可以发现，子空间迭代法把一个 n 维的广义特征值问题化成了 q 维的广义特征值问题，一般情况下 $q \ll n$，因此计算工作量可以大大减少。

在 MATLAB 中，其实已经提供了专门的函数来计算矩阵的特征值，即 eig 和 eigs。它们既可用于计算标准特征值问题，也可以用于求解广义特征值问题。前者一般用来求解维数较小时的全部特征值，而后者一般用来获取大型特征值问题的某几个特征对，这在有限元中非常常见。因为结构的最低几阶频率往往是设计人员最感兴趣的，而且随频率的增高，有限元的误差会越来越大。如果我们已经有了整体的刚度矩阵 K 和质量矩阵 M，那么用 eigs 来计算最低几阶频率的工作在 MATLAB 中就只要用一句语句就可以，如：

```
[EigVector, EigValue] = eigs(K, M, 6, 'SM');
```

上面这句语句就能得到前 6 阶特征值和特征向量。这里要注意的是，该函数要求矩阵 $\boldsymbol{M}$ 必须是对称正定的，这一点在程序中要特别注意。因此非正定的集中质量阵不能用它。另外由于计算中的数值误差会导致质量矩阵 $\boldsymbol{M}$ 不对称，解决的办法是在调用 eigs 函数前，把质量矩阵 $\boldsymbol{M}$ 的上三角元素赋值给相应的下三角元素，使它严格对称。

8.4 结构的稳定

结构的线性稳定问题，可最终归结为一个特征值问题。

有限元法对结构进行线性稳定分析通常要分两步，首先用前面介绍的线性分析方法求出结构的内力分布，然后根据内力分布计算几何刚度矩阵，与刚度矩阵一起构成广义特征值问题。这样得到的特征值和特征向量即为临界荷载及其相应的屈曲形态。

这里讨论的是所谓线弹性稳定问题，即结构的内力由线弹性分析决定，而且在屈曲引起的无限小位移过程中，外力保持不变。至于非线性屈曲或屈曲后的形态，将是非线性大变形问题，不在本书讨论。

8.4.1 杆的稳定

杆的临界荷载是使杆在没有横向力作用下保持弯曲平衡的轴向力，所以这一问题实际上是轴向力作用下梁的弯曲问题。下面以平面梁单元为例来说明。梁的平面弯曲问题已经在第 3 章讨论过。现在，设某单元的轴向力为 p，于是要计算轴向力 p 所引起的梁单元的等效结点力。

在第 3 章中，平面梁单元的弯曲挠度插值公式为

$$v = \boldsymbol{H}_2(x)\boldsymbol{A}_2^{-1}\boldsymbol{\delta}_v \tag{8-66}$$

转角是挠度 v 对 x 的一阶导数，因此梁内任一点的位移向量为

$$\begin{Bmatrix} v \\ \theta \end{Bmatrix} = \begin{bmatrix} \boldsymbol{H}_2(x) \\ \boldsymbol{H}_2'(x) \end{bmatrix}\boldsymbol{A}_2^{-1}\boldsymbol{\delta}_v \tag{8-67}$$

与此相应，轴向力的广义体积力分量为

$$\boldsymbol{p}_v = \begin{Bmatrix} 0 \\ -pv' \end{Bmatrix} \tag{8-68}$$

把上式和现在的形函数表达式

$$N=\begin{bmatrix}\mathbf{H}_2(x)\\ H_2'(x)\end{bmatrix}A_2^{-1}\tag{8-69}$$

代入等效结点力公式

$$f_v^e=\int_0^l N^T p_t\,dx\tag{8-70}$$

得到

$$f_v^e=-K_g^e\delta_v\tag{8-71}$$

式中 K_g^e 称为单元的几何刚度矩阵

$$K_g^e=A_2^{-T}\int_0^l H_2'(x)^T p H_2'(x)\,dx A_2^{-1}\tag{8-72}$$

由于不考虑梁的轴向位移时，而外力又只有轴向力时，梁单元的平衡方程为

$$K^e\delta_v=f_v^e\tag{8-73}$$

所以把(8-71)代入上式,得到

$$(K^e+K_g^e)\delta_v=0\tag{8-74}$$

单元几何刚度矩阵可以按单元刚度矩阵一样的方式组集成整体几何刚度矩阵，因此可以得到整体平衡方程

$$(K+K_g)\delta=0\tag{8-75}$$

式中 K_g 称为整体几何刚度矩阵，它是对称的，元素分布与刚度矩阵一样。当单元中的轴力 p 是常量时,由式(8-72)得

$$K_g^e=\frac{p}{30l}\begin{bmatrix}36 & 3l & -36 & 3l\\ 3l & 4l^2 & -3l & -l^2\\ -36 & -3l & 36 & -3l\\ 3l & -l^2 & -3l & 4l^2\end{bmatrix}\tag{8-76}$$

一般来说，方程式(8-75)的系数矩阵是非奇异的,它只有零解,表示原来的非挠曲的平衡是稳定的。设外力按比例增加 λ 倍,单元轴力变成 λp,单元的和整体的几何刚度矩阵分别变为 λK_g^e 和 λK_g。总体的平衡方程为

$$(K+\lambda K_g)\delta=0\tag{8-77}$$

我们发现上式与式(8-27)在形式上是一致的，即结构的稳定问题与结构的模态分析在数学上都是广义特征值问题。在某些 λ 值，上式的系数矩阵变为奇异，方程有非零解，表示挠曲形式也是平衡位置。此时如果有微小的横向扰动，弯曲位移会变成无穷大。

若式(8-77)为 n 阶,便有 n 个特征对:λ_i 和 $\varphi_i(i=1,2,\cdots,n)$。相应的

外荷载 $\lambda_i p$ 便是临界荷载。φ_i 便是失稳时的屈曲形式。实际上，只有最小的正特征值所对应的临界荷载才有意义。如果特征方程（8-77）没有正特征值，说明在这种荷载下，结构没有失稳问题。例如，杆在轴向拉力作用下，不存在失稳问题。

8.4.2 板的稳定

板的稳定问题可以从杆的稳定问题推广得到。此时，板的面内应力由平面应力问题求得，设为 σ_x σ_y 和 τ_{xy}。板内任意一点的挠度插值公式

$$w = N\delta^e \tag{8-78}$$

板内任意一点的位移向量为

$$u = \begin{Bmatrix} w \\ \theta_x \\ \theta_y \end{Bmatrix} = \begin{Bmatrix} w \\ \partial w/\partial y \\ -\partial w/\partial x \end{Bmatrix} = \begin{bmatrix} N \\ N_{,y} \\ -N_{,x} \end{bmatrix}\delta^e \tag{8-79}$$

与此相应，板面内应力的广义体积力分量是

$$p_v = \begin{Bmatrix} 0 \\ -(\sigma_y \partial w/\partial y + \tau_{xy}\partial w/\partial x) \\ \tau_{xy}\partial w/\partial y + \sigma_x \partial w/\partial x \end{Bmatrix} \tag{8-80}$$

因此广义体积力的等效结点力是

$$f_v^e = -\left(\iint_\Omega h\begin{bmatrix} N_{,x}^{\mathrm{T}} & N_{,y}^{\mathrm{T}} \end{bmatrix}\begin{bmatrix} \sigma_x & \tau_{xy} \\ \tau_{xy} & \sigma_y \end{bmatrix}\begin{bmatrix} N_{,x} \\ N_{,y} \end{bmatrix}\mathrm{d}x\mathrm{d}y\right)\delta^e \tag{8-81}$$

因此单元的几何刚度矩阵是

$$K_g^e = \sigma_x K_{gx}^e + \sigma_y K_{gy}^e + \tau_{xy} K_{gxy}^e \tag{8-82}$$

式中

$$K_{gx}^e = \iint_\Omega hN_{,x}^{\mathrm{T}}N_{,x}\mathrm{d}x\mathrm{d}y \tag{8-83}$$

$$K_{gy}^e = \iint_\Omega hN_{,y}^{\mathrm{T}}N_{,y}\,\mathrm{d}x\mathrm{d}y \tag{8-84}$$

$$K_{gxy}^e = 2\iint_\Omega hN_{,x}^{\mathrm{T}}N_{,y}\,\mathrm{d}x\mathrm{d}y \tag{8-85}$$

有了板单元的几何刚度矩阵，可以与杆的稳定一样，经过组集得到形式上完全一样的整体平衡方程，求解相应的特征值问题，即可得到板的临界荷载。

8.5 结构的动力响应

在第一节，已经得到的结构动力学方程式为

$$\boldsymbol{M}\ddot{\boldsymbol{\delta}} + \boldsymbol{C}\dot{\boldsymbol{\delta}} + \boldsymbol{K}\boldsymbol{\delta} = \boldsymbol{f} \tag{8-86}$$

不妨设它已经做了边界约束处理。位移向量 $\boldsymbol{\delta}$ 是时间 t 的函数，速度向量 $\dot{\boldsymbol{\delta}}$ 和加速度向量 $\ddot{\boldsymbol{\delta}}$ 分别是位移向量对时间的一阶导数和二阶导数，载荷向量 $\boldsymbol{f}$ 是时间的已知函数。方程式（8-86）是二阶常微分方程组。它应有两个初始条件，就是结构初始时刻的位移和速度，设为

$$\boldsymbol{\delta} = \boldsymbol{\delta}_0, \quad \dot{\boldsymbol{\delta}} = \dot{\boldsymbol{\delta}}_0 \qquad \text{当 } t = 0 \text{ 时} \tag{8-87}$$

若 $\boldsymbol{f}$ 为时间 t 的周期函数，例如 $\boldsymbol{f} = \boldsymbol{f}\sin(\omega t)$，由于阻尼的作用，初始条件（8-87）所引起的振动将衰减趋近于零，因此只需考虑强迫力所激起的稳态振动。求稳态振动只需设

$$\boldsymbol{\delta} = \boldsymbol{\delta}\sin(\omega t) \tag{8-88}$$

代入方程式（8-86），得到代数方程组

$$(-\omega^2 \boldsymbol{M} + \omega \boldsymbol{C} + \boldsymbol{K})\boldsymbol{\delta} = \boldsymbol{f} \tag{8-89}$$

解之便得到 $\boldsymbol{\delta}$。对于 $\boldsymbol{f}$ 是其他周期函数的情况，将 $\boldsymbol{f}$ 展开成 Fourier 级数也不难求解。但是对于非周期函数，如冲击载荷，或只考虑结构在某时间历程内的响应问题，上述方法不再适用。求解此问题的方法很多，常用的有振型迭加法加以及直接积分法，或称逐步积分法。振型叠加法只适用于线性系统，逐步积分法也适用于非线性系统。

8.5.1 振型叠加法

振型叠加法的基本思想是，利用结构自由振动的振型相互正交的特性，将结构的动力学方程化成各广义坐标的非耦合方程，然后各方程可用 Duhamel 积分或其他方法单独求解。求结构的自由振动振型，可按前章讲的步骤进行。在阻尼自由振动的方程

$$\boldsymbol{M}\boldsymbol{\delta} + \boldsymbol{C}\boldsymbol{\delta} + \boldsymbol{K}\boldsymbol{\delta} = \boldsymbol{0} \tag{8-90}$$

中，代入如下形式的解

$$\boldsymbol{\delta} = \boldsymbol{\delta} e^{\omega t} \tag{8-91}$$

得特征方程

$$(\boldsymbol{M}\omega^2 + \boldsymbol{C}\omega + \boldsymbol{K})\bar{\boldsymbol{\delta}} = \boldsymbol{0} \tag{8-92}$$

式中 ω 是复参数。为了方便，上式不妨记做

$$(\boldsymbol{M}\omega^2+\boldsymbol{C}\omega+\boldsymbol{K})\boldsymbol{\delta}=\boldsymbol{0} \tag{8-93}$$

此特征值问题的特征值 ω_i 和特征向量 $\boldsymbol{\varphi}_i$ 均为复数。可以证明，特征向量仍具有正交性：

$$\begin{aligned}\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{M}\boldsymbol{\varphi}_j&=\delta_{ij}\\ \boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{C}\boldsymbol{\varphi}_j&=2\xi_i\,\omega_i\,\delta_{ij}\\ \boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\varphi}_j&=\omega_i^2\delta_{ij}\end{aligned} \tag{8-94}$$

式中，ξ_i 为结构第 i 阶振型阻尼比，它一般由试验确定。注意这里的运算应为复向量运算，矩阵 $\boldsymbol{A}$ 的转置应取

$$\boldsymbol{A}^{\mathrm{T}}=\bar{\boldsymbol{A}}^{\mathrm{T}} \tag{8-95}$$

式中，矩阵 $\bar{\boldsymbol{A}}$ 表示矩阵 $\boldsymbol{A}$ 的复共轭。先假定 n 阶特征值问题式(8-92)有 n 个特征向量组成完备的向量基，并记

$$\begin{aligned}\boldsymbol{\Phi}&=[\boldsymbol{\varphi}_1\quad\boldsymbol{\varphi}_2\quad\cdots\quad\boldsymbol{\varphi}_n]\\ \boldsymbol{\Omega}&=\mathrm{diag}[\omega_1\quad\omega_2\quad\cdots\quad\omega_n]\\ \boldsymbol{\Xi}&=\mathrm{diag}[\xi_1\quad\xi_2\quad\cdots\quad\xi_n]\end{aligned} \tag{8-96}$$

作变换

$$\boldsymbol{\delta}=\boldsymbol{\Phi}\boldsymbol{x} \tag{8-97}$$

将方程式(8-86)化成

$$\ddot{\boldsymbol{x}}+2\boldsymbol{\Xi}\boldsymbol{\Omega}\dot{\boldsymbol{x}}+\boldsymbol{\Omega}^2\boldsymbol{x}=\boldsymbol{g} \tag{8-98}$$

式中

$$\boldsymbol{g}=\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{f} \tag{8-99}$$

将初始条件化为

$$\left.\begin{aligned}\boldsymbol{x}_0&=\boldsymbol{\Phi}^{-1}\boldsymbol{\delta}_0\\ \dot{\boldsymbol{x}}_0&=\boldsymbol{\Phi}^{-1}\dot{\boldsymbol{\delta}}_0\end{aligned}\right\}\quad 当\ t=0\ 时 \tag{8-100}$$

式(8-98)是一组非耦合方程，即 $x_i(i=1,2\cdots n)$ 之间不耦合，可以单独求解，把它写成分量的形式，便是

$$\left.\begin{aligned}&\ddot{x}_i+2\xi_i\,\omega_i\dot{x}_i+\omega_i^2x_i=g_i\\ &x_i\big|_{t=0}=x_{i0},\ \dot{x}_i\big|_{t=0}=\dot{x}_{i0}\end{aligned}\right\}\quad (i=1,2,\cdots,n) \tag{8-101}$$

上式的解可以用 Duhamel 积分表示

$$\begin{aligned}x_i=&\frac{1}{\omega'_i}\int_0^t g_i(\tau)\exp[-\xi_i\omega_i(t-\tau)]\sin[\omega_i(t-\tau)]\mathrm{d}\tau\\ &+\exp(-\xi_i\omega_i t)[\alpha_i\sin(\omega'_i t)+\beta_i\cos(\omega'_i t)]\end{aligned} \tag{8-102}$$

式中，$\omega'_i = \omega_i\sqrt{1-\xi_i^2}$，$\alpha_i$ 和 β_i 由初始条件定出。得到了 $\boldsymbol{x}$ 后，再代入式(8-97)即可得到 $\boldsymbol{\delta}$。

求解阻尼自由振动问题式(8-93)，要求解复特征值问题，计算工作量很大。那么用无阻尼自由振动的固有频率作为向量基作变换，能不能同样达到将式(8-86)解耦的目的呢？问题的关键在阻尼矩阵。如果无阻尼自由振动的特征向量也能关于 $\boldsymbol{C}$ 正交，即

$$\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{\Phi} = 2\boldsymbol{\Xi}\boldsymbol{\Omega} \tag{8-103}$$

则用无阻尼自由振动的特征向量矩阵 $\boldsymbol{\Phi}$ 作变换矩阵，也能将方程式(8-86)解耦为式(8-101)。所以，为简单起见，一般构造阻尼矩阵 $\boldsymbol{C}$ 都希望满足式(8-103)。最简单的就是前面所说的 Rayleigh 阻尼，它取为质量矩阵和刚度矩阵的线性组合

$$\boldsymbol{C} = \alpha\boldsymbol{M} + \beta\boldsymbol{K} \tag{8-104}$$

式中 α、β 为待定常数，由实验确定。将上式代入(8-103)，得到

$$\alpha + \beta\boldsymbol{\Omega}^2 = 2\boldsymbol{\Xi}\boldsymbol{\Omega} \tag{8-105}$$

即为

$$\alpha + \beta\omega_i^2 = 2\xi_i\omega_i \quad (i = 1,2,\cdots,n) \tag{8-106}$$

一般用两阶振型阻尼比，例如最低两阶振型阻尼比 ξ_1 和 ξ_2(一般由实验给出)，便可定出 α 和 β，其他高阶的振型阻尼比也随之确定

$$\xi_i = \frac{\alpha + \beta\omega_i^2}{2\omega_i} \qquad (i = 3,4,\cdots) \tag{8-107}$$

由此可以看出，Rayleigh 阻尼中高阶阻尼比很高。这实际上就等于在计算中将高阶振型的响应消除了。因此振型叠加法适用于像地震等只激发起较少振型，所需计算的响应历程较长这类问题。由于高阶阻尼比较高，计算时不必考虑全部振型，只需要取少数最低几阶振型即可。而对于像冲击等问题，激发起的振型较多，所需计算响应的历程短，通常用逐步积分法为宜。

8.5.2 逐步积分法

逐步积分法根据动力学方程(8-86)，引进某些假设，建立由 t 时刻状态向量 $\boldsymbol{\delta}_t$、$\dot{\boldsymbol{\delta}}_t$ 和 $\ddot{\boldsymbol{\delta}}_t$ 到 $t+\Delta t$ 时刻状态向量 $\boldsymbol{\delta}_{t+\Delta t}$、$\dot{\boldsymbol{\delta}}_{t+\Delta t}$ 和 $\ddot{\boldsymbol{\delta}}_{t+\Delta t}$ 的递推关系，从而从 $t=0$ 时刻的初始状态向量 $\boldsymbol{\delta}_0$、$\dot{\boldsymbol{\delta}}_0$ 和 $\ddot{\boldsymbol{\delta}}_0$ 出发，逐步地求出各时刻的状态向量。在初始时刻状态向量中，初始加速度 $\ddot{\boldsymbol{\delta}}_0$ 可在平衡方程(8-86)中直接代入初始条件式(8-87)求得。由于所取的假设不同，可以有各种各样的方法。这里介绍

常用的 Newmark 法和 Wilson—θ 法。逐步积分法不仅适用于以物理坐标表示的方程组(8-86)，也可用于以主振型表示的非耦合方程式(8-101)，以代替 Duhamel 积分。

1. Newmark 法

建立逐步积分法的关键是建立由 t 时刻到 $t+\Delta t$ 时刻的状态向量的递推关系。$t+\Delta t$ 时刻有 3 组未知量 $\boldsymbol{\delta}_{t+\Delta t}$，$\dot{\boldsymbol{\delta}}_{t+\Delta t}$和 $\ddot{\boldsymbol{\delta}}_{t+\Delta t}$，它们满足动力学方程式(8-86)，即

$$\boldsymbol{M}\ddot{\boldsymbol{\delta}}_{t+\Delta t} + \boldsymbol{C}\dot{\boldsymbol{\delta}}_{t+\Delta t} + \boldsymbol{K}\boldsymbol{\delta}_{t+\Delta t} = \boldsymbol{f}_{t+\Delta t} \tag{8-108}$$

显然还应补充二组方程才能求解。Newmark 法假设 $t+\Delta t$ 时刻的速度和位移为

$$\dot{\boldsymbol{\delta}}_{t+\Delta t} = \dot{\boldsymbol{\delta}}_t + [(1-\gamma)\ddot{\boldsymbol{\delta}}_t + \gamma\ddot{\boldsymbol{\delta}}_{t+\Delta t}]\Delta t \quad 0 \leqslant \gamma \leqslant 1 \tag{8-109}$$

$$\boldsymbol{\delta}_{t+\Delta t} = \boldsymbol{\delta}_t + \dot{\boldsymbol{\delta}}_t\Delta t + \left[\left(\frac{1}{2}-\beta\right)\ddot{\boldsymbol{\delta}}_t + \beta\ddot{\boldsymbol{\delta}}_{t+\Delta t}\right]\Delta t^2 \quad 0 \leqslant 2\beta \leqslant 1 \tag{8-110}$$

式(8-108)～式(8-110)是 Newmark 法的基本公式。实际计算中，可以先从式(8-110)得到用 $\boldsymbol{\delta}_{t+\Delta t}$表示的 $\ddot{\boldsymbol{\delta}}_{t+\Delta t}$，再代入(8-109)得到用 $\boldsymbol{\delta}_{t+\Delta t}$表示的 $\dot{\boldsymbol{\delta}}_{t+\Delta t}$。把它们全部代入到公式(8-108)，就得到一个未知量只有 $\boldsymbol{\delta}_{t+\Delta t}$的方程组，求解它即能得到 $t+\Delta t$ 时刻的位移向量 $\boldsymbol{\delta}_{t+\Delta t}$。再把它代回到(8-109)和式(8-110)可得到 $t+\Delta t$ 时刻的速度 $\dot{\boldsymbol{\delta}}_{t+\Delta t}$和加速度 $\ddot{\boldsymbol{\delta}}_{t+\Delta t}$。

参数 γ 和 β的选择对算法影响较大。算法稳定性分析指出，当 $\gamma \geqslant 0.5$，$\beta \geqslant \left(\frac{1}{2}+\gamma\right)^2/4$时，Newmark 法无条件稳定，这时可以只根据精度的要求选择时间步长 Δt。

最后综合计算步骤如下[6]：

A. 初始计算

(A.1)形成刚度 $\boldsymbol{K}$，质量矩阵 $\boldsymbol{M}$和阻尼矩阵 $\boldsymbol{C}$。

(A.2)获得初始状态向量 $\boldsymbol{\delta}_0$、$\dot{\boldsymbol{\delta}}_0$ 和 $\ddot{\boldsymbol{\delta}}_0$

(A.3)选择时间步长 Δt 以及参数 γ 和 β，并计算下列有关常数

$$\alpha_0 = \frac{1}{\beta\Delta t^2},\quad \alpha_1 = \frac{\gamma}{\beta\Delta t},\quad \alpha_2 = \frac{1}{\beta\Delta t},\quad \alpha_3 = \frac{1}{2\beta} - 1$$

$$\alpha_4 = \frac{\gamma}{\beta} - 1,\quad \alpha_5 = \frac{\Delta t}{2}\left(\frac{\gamma}{\beta} - 2\right),\quad \alpha_6 = \Delta t(1-\gamma),\quad \alpha_7 = \gamma\Delta t$$

(A.4)计算有效刚度矩阵

$$\hat{\boldsymbol{K}} = \boldsymbol{K} + \alpha_0\boldsymbol{M} + \alpha_1\boldsymbol{C}$$

B. 对每个时间步计算

(B.1) 计算 $t+\Delta t$ 时刻的有效载荷向量

$$\tilde{\boldsymbol{f}}_{t+\Delta t}=\boldsymbol{f}_{t+\Delta t}+\boldsymbol{M}(\alpha_0\boldsymbol{\delta}_t+\alpha_2\dot{\boldsymbol{\delta}}_t+\alpha_3\ddot{\boldsymbol{\delta}}_t)+\boldsymbol{C}(\alpha_1\boldsymbol{\delta}_t+\alpha_4\dot{\boldsymbol{\delta}}_t+\alpha_5\ddot{\boldsymbol{\delta}}_t)$$

(B.2) 求 $t+\Delta t$ 时刻的位移

$$\tilde{\boldsymbol{K}}\boldsymbol{\delta}_{t+\Delta t}=\tilde{\boldsymbol{f}}_{t+\Delta t}$$

(B.3) 计算 $t+\Delta t$ 时刻的加速度和速度

$$\ddot{\boldsymbol{\delta}}_{t+\Delta t}=\alpha_0(\boldsymbol{\delta}_{t+\Delta t}-\boldsymbol{\delta}_t)-\alpha_2\dot{\boldsymbol{\delta}}_t-\alpha_3\ddot{\boldsymbol{\delta}}_t$$

$$\dot{\boldsymbol{\delta}}_{t+\Delta t}=\dot{\boldsymbol{\delta}}_t+\alpha_6\ddot{\boldsymbol{\delta}}_t+\alpha_7\ddot{\boldsymbol{\delta}}_{t+\Delta t}$$

这里需要指出的是，在步骤(B.2)中计算 $t+\Delta t$ 时刻的位移时，需要求解的线性方程组的刚度矩阵 $\tilde{\boldsymbol{K}}$ 是不随时间变化的，因此，可以预先对其进行三角分解，在每个时间步只需要回代即可。下面给出了用 MATLAB 编写的 Newmark 法的程序段，但是并没有预先对刚度矩阵进行三角分解，这样可使程序更加简单易懂。

```
1  [illegible]
2  [illegible]
3  [illegible]
4  [illegible]
5  [illegible]
6  %  输入参数
7  %     K  ---- 刚度矩阵
8  %     M  ---- 质量矩阵
   %     C  ---- 阻尼矩阵
   %     d0 ---- 初始位移
   %     v0 ---- 初始速度
10 %     dt ---- 时间步长
11 %     tend ---- 结束时间
12 %  返回值
13 %     d  ---- 位移
14 %     v  ---- 速度
15 %     a  ---- 加速度
16     gama = 0.5;
17     beta = 0.25;
18     [illegible] = size(K);
19     alpha0 = 1/beta/dt^2;
20     alpha1 = gama/beta/dt;
21     alpha2 = 1/beta/dt;
22     alpha3 = 1/2/beta - 1;
```

```
23.     alpha4 = gama/beta - 1 ;
24.     alpha5 = dt/2*(gama/beta-2) ;
25.     [illegible] dt (1-gama) ;
26.     alpha7 = gama*dt ;
27.     K1 = K + alpha0*M + alpha1*C ;
28.     d = zeros(n, floor(tend/dt) + 1) ;
29.     v = zeros(n, floor(tend/dt) + 1) ;
30.     a = zeros(n, floor(tend/dt) + 1) ;
31.     d(:,1) = d1 ;
32.     v(:,1) = v1 ;
33.     a(:,1) = M\(f(:,1) - K*d1 - C*v1) ;
34.     for i = 2:1:floor(tend/dt) + 1
35.         t = (i-1)*dt ;
36.         f1 = f(:,i) + M*[illegible]
37.         [illegible]
38.         [illegible]
39.         [illegible]
40.         v(:,i) = v(:,i-1) + alpha6*a(:,i-1) [illegible]
41.     end
42. return
```

2. Wilson—θ 法

Wilson—θ法是线性加速度法的推广。它的基本假定是加速度在 $[t, t+\theta\Delta t]$ 范围内线性变化,即

$$\ddot{\boldsymbol{\delta}}_{t+\tau} = \ddot{\boldsymbol{\delta}}_t + \frac{\tau}{\theta\Delta t}(\ddot{\boldsymbol{\delta}}_{t+\theta\Delta t} - \ddot{\boldsymbol{\delta}}_t) \quad 0 \leqslant \tau \leqslant \theta\Delta t \tag{8-111}$$

对上式积分两次,分别得到

$$\dot{\boldsymbol{\delta}}_{t+\tau} = \dot{\boldsymbol{\delta}}_t + \ddot{\boldsymbol{\delta}}_t \tau + \frac{\tau^2}{2\theta\Delta t}(\ddot{\boldsymbol{\delta}}_{t+\theta\Delta t} - \ddot{\boldsymbol{\delta}}_t) \quad 0 \leqslant \tau \leqslant \theta\Delta t \tag{8-112}$$

和

$$\boldsymbol{\delta}_{t+\tau} = \boldsymbol{\delta}_t + \dot{\boldsymbol{\delta}}_t \tau + \frac{1}{2}\ddot{\boldsymbol{\delta}}_t \tau^2 + \frac{\tau^3}{6\theta\Delta t}(\ddot{\boldsymbol{\delta}}_{t+\theta\Delta t} - \ddot{\boldsymbol{\delta}}_t) \quad 0 \leqslant \tau \leqslant \theta\Delta t \tag{8-113}$$

这样在 $t+\theta\Delta t$ 时刻的速度和位移为

$$\dot{\boldsymbol{\delta}}_{t+\theta\Delta t} = \dot{\boldsymbol{\delta}}_t + \frac{\theta\Delta t}{2}(\ddot{\boldsymbol{\delta}}_{t+\theta\Delta t} + \ddot{\boldsymbol{\delta}}_t) \tag{8-114}$$

$$\boldsymbol{\delta}_{t+\theta\Delta t}=\boldsymbol{\delta}_t+\theta\Delta t\dot{\boldsymbol{\delta}}_t+\frac{\theta^2\Delta t^2}{6}(\ddot{\boldsymbol{\delta}}_{t+\theta\Delta t}+2\ddot{\boldsymbol{\delta}}_t) \tag{8-115}$$

从上面两式，可以把 $t+\theta\Delta t$ 时刻的加速度 $\ddot{\boldsymbol{\delta}}_{t+\theta\Delta t}$和速度 $\dot{\boldsymbol{\delta}}_{t+\theta\Delta t}$用 $t+\theta\Delta t$ 时刻的位移 $\boldsymbol{\delta}_{t+\theta\Delta t}$来表示，即

$$\ddot{\boldsymbol{\delta}}_{t+\theta\Delta t}=\frac{6}{\theta^2\Delta t^2}(\boldsymbol{\delta}_{t+\theta\Delta t}-\boldsymbol{\delta}_t)-\frac{6}{\theta\Delta t}\dot{\boldsymbol{\delta}}_t-2\ddot{\boldsymbol{\delta}}_t \tag{8-116}$$

$$\dot{\boldsymbol{\delta}}_{t+\theta\Delta t}=\frac{3}{\theta\Delta t}(\boldsymbol{\delta}_{t+\theta\Delta t}-\boldsymbol{\delta}_t)-2\dot{\boldsymbol{\delta}}_t-\frac{\theta\Delta t}{2}\ddot{\boldsymbol{\delta}}_t \tag{8-117}$$

考虑在 $t+\theta\Delta t$ 时刻的动力学方程(8-86)为

$$\boldsymbol{M}\ddot{\boldsymbol{\delta}}_{t+\theta\Delta t}+\boldsymbol{C}\dot{\boldsymbol{\delta}}_{t+\theta\Delta t}+\boldsymbol{K}\boldsymbol{\delta}_{t+\theta\Delta t}=\boldsymbol{f}_{t+\theta\Delta t} \tag{8-118}$$

因为我们假设加速度是在$[t,t+\theta\Delta t]$内是线性变化的，因此载荷也应是在$[t,t+\theta\Delta t]$内是线性变化的，即

$$\boldsymbol{f}_{t+\theta\Delta t}=\boldsymbol{f}_t+\theta(\boldsymbol{f}_{t+\Delta t}-\boldsymbol{f}_t) \tag{8-119}$$

把式(8-116)、式(8-117)和式(8-119)代入式(8-118)即可求得 $t+\theta\Delta t$ 的位移 $\boldsymbol{\delta}_{t+\theta\Delta t}$，把它代回到式(8-116)和式(8-117)，就能得到 $t+\theta\Delta t$ 时刻的加速度 $\ddot{\boldsymbol{\delta}}_{t+\theta\Delta t}$和速度 $\dot{\boldsymbol{\delta}}_{t+\theta\Delta t}$。再应用式(8-111)～式(8-113)，并令 $\tau=\Delta t$，即得到 $t+\Delta t$ 的位移、速度和加速度。

同样由算法稳定性分析指出，当 $\theta>1.366$ 时，该法是无条件稳定的，计算标明一般取 $\theta=1.40$ 为宜。计算还标明 Wilson 法有较高的算法阻尼，即由于算法引进的所谓阻尼。对于结构的高阶振型算法阻尼尤高，这就取消了高阶振型对响应的贡献。由于有限元法离散必然引起较大的高阶振型的误差，考虑高阶振型没有什么意义，通过积分过程的算法阻尼，把高阶振型的响应消除掉是合理的。

Wilson-θ法的计算步骤如下[6]。

A. 初始计算

(A.1) 形成刚度 $\boldsymbol{K}$，质量矩阵 $\boldsymbol{M}$ 和阻尼矩阵 $\boldsymbol{C}$。

(A.2) 获得初始状态向量 $\boldsymbol{\delta}_0$、$\dot{\boldsymbol{\delta}}_0$ 和 $\ddot{\boldsymbol{\delta}}_0$。

(A.3) 选择时间步长 Δt，并计算积分常数

$$\theta=1.4,\quad \tau=\theta\Delta t,\quad \alpha_0=\frac{6}{\tau^2},\quad \alpha_1=\frac{3}{\tau},\quad \alpha_2=2\alpha_1,\quad \alpha_3=\frac{\tau}{2},$$

$$\alpha_4=\frac{\alpha_0}{\theta},\quad \alpha_5=-\frac{\alpha_2}{\theta},\quad \alpha_6=1-\frac{3}{\theta},\quad \alpha_7=\frac{\Delta t}{2},\quad \alpha_8=\frac{\Delta t^2}{6}$$

(A.4) 形成有效刚度矩阵

$$\tilde{K} = K + \alpha_0 M + \alpha_1 C$$

B.对每个时间步计算

(B.1)计算 $t+\theta\Delta t$ 时刻的有效载荷向量

$$\hat{f}_{t+\theta\Delta t} = f_{t+\theta\Delta t} + M(\alpha_0 \delta_t + \alpha_2 \dot{\delta}_t + 2\ddot{\delta}_t) + C(\alpha_1 \delta_t + 2\dot{\delta}_t + \alpha_3 \ddot{\delta}_t)$$

(B.2)求 $t+\theta\Delta t$ 时刻的位移

$$\tilde{K}\delta_{t+\theta\Delta t} = \hat{f}_{t+\theta\Delta t}$$

(B.3)计算 $t+\Delta t$ 时刻的加速度、速度和位移

$$\ddot{\delta}_{t+\Delta t} = \alpha_4(\delta_{t+\theta\Delta t} - \delta_t) + \alpha_5 \dot{\delta}_t + \alpha_6 \ddot{\delta}_t$$

$$\dot{\delta}_{t+\Delta t} = \dot{\delta}_t + \alpha_7(\ddot{\delta}_{t+\Delta t} + \ddot{\delta}_t)$$

$$\delta_{t+\Delta t} = \delta_t + \Delta t\dot{\delta}_t + \alpha_8(\ddot{\delta}_{t+\Delta t} + 2\ddot{\delta}_t)$$

同样,下面也给出了 Wilson—θ 法的程序,以有助于理解上面的公式

```
1.  [illegible]
2.  [illegible]
3.  [illegible]
4.  [illegible]
5.  [illegible]
6.  [illegible]
7.  [illegible]
8.  [illegible]
9.  [illegible]
10. [illegible]
11. [illegible]
12. [illegible]
13. %    d  ------ 位移
14. %    v  ------ 速度
15. %    a  ------ 加速度
16.      [n,n] = size(K) ;
17.      theta = 1.4 ;
18.      tao = theta* dt ;
19.      alpha0 = 6/tao^2 ;
20.      alpha1 = 3/tao ;
21.      alpha2 = [illegible]
22.      alpha3 = [illegible]
23.      alpha4 = alpha0/theta [illegible]
```

```
24.     alpha5 = -alpha2/theta ;
25.     alpha6 = 1 -3/theta ;
26.     alpha7 = dt/2 ;
27.     alpha8 = dt^2/6 ;
28.     K1 [illegible] alpha1*C [illegible]
29.     d = zeros( n, floor(tend/dt) + 1 ) ;
30.     v = zeros( n, floor(tend/dt) + 1 ) ;
31.     a = zeros( n, floor(tend/dt) + 1 ) ;
32.     d(:,1) = d1 ;
33.     v(:,1) = v1 ;
34.     a(:,1) = M\(f(:,1) - K*d1 - C*v1) ;
35.     for i =2:1:floor(tend/dt)
36.         t = (i-1)*dt ;
37.         [illegible]
38.         fq = f(:,i-1 [illegible]
                 [illegible]
39.         f1 = fq + M*(alpha0*d(:,i-1) [illegible]
40.                 + C*(alpha1*d(:,i-1) [illegible]
41.         dq = K1\f1 ;
42.         a(:,i) = alpha4*(dq-d(:,i-1)) [illegible]
                     + alpha6*a(:,i-1) [illegible]
43.         v(:,i) = v(:,i-1) + alpha7*(a(:,i) [illegible]
44.         d(:,i) = d(:,i-1) + dt*v(:,i-1) + alpha8*(a(:,i)
                     +2*a(:,i-1)) ;
45.     end
46. return
```

8.6 算例

8.6.1 两铰抛物线拱的面内自由振动

如图 8-1 所示，已知跨径 $L = 60$ m，矢高 $f = 7.5$m，弹性模量 $E = 206$GPa，拱截面积 $A = 0.0815\text{m}^2$，抗弯惯性矩 $I = 0.03622\text{m}^4$，线密度 $\rho = 1435.2$ kg/m。采用 100个平面梁单元，得到前三阶固有频率和振型如图 8-2 所示。所用的 MATLAB 程序见本书所附文件 Exam8_1.m

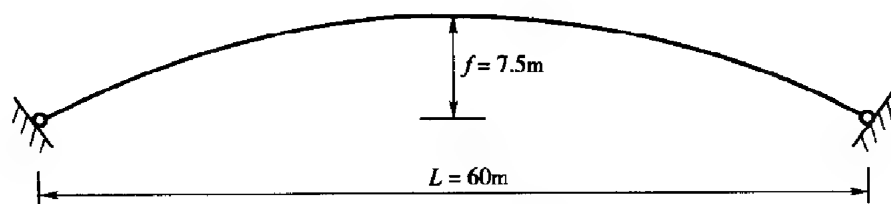


图 8-1　两铰抛物线拱

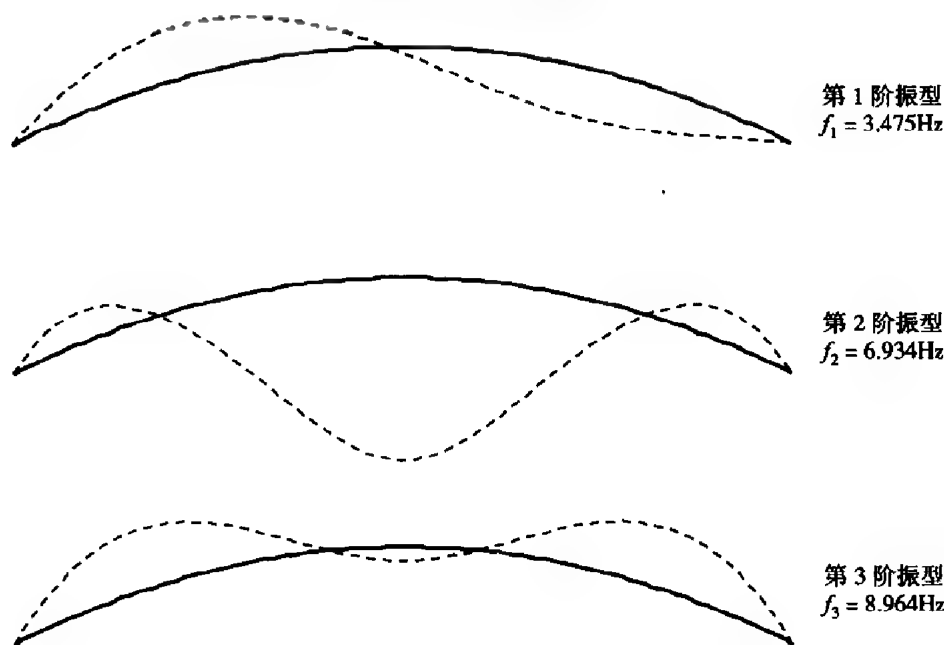


图 8-2　两铰抛物线拱的前三阶面内振动频率和振型

8.6.2　抛物线拱的时程响应分析

计算对象仍然是上述算例中的两铰拱，分析它在初速度下的自由振动。计算时，设所有结点的初位移为零，而初速度均为 1m/s 。对计算所得的 $L/4$ 跨处结点的位移时程曲线进行 FFT 变换，得到其频谱特性和前三阶的自由振动频率，如图 8-3 和图 8-4 所示。计算时采用 Newmark 法，取 $\gamma=0.5$，$\beta=0.25$，时间步长为 $\Delta t=0.01\text{s}$，计算的时间长度为 $t=\Delta t\times 4096=40.96\text{s}$，FFT 变换的频率分辨率为 $\Delta f=1/t=0.0244\text{Hz}$。

图 8-4 显示，第 1 阶频率和第 3 阶频率的振幅比第 2 阶频率的振幅要大许多，这可以从图 8-2 的三阶模态的振型中得到印证，即在 $L/4$ 跨处第 2 阶模态的位移最小。如果用跨中挠度的时程曲线进行频谱分析，就会发现得不到第 1 阶频率，这是因为对于第 1 阶频率，跨中处是节点，该处的挠度永远为零。另外还可以看出按时程曲线进行频谱分析所得的频率与模态分析方法所得频

率为小。这是因为用 Newmark 法进行时程分析时会导致周期伸长[6]，即频率变小。本算例源程序见本书所附文件 exam8_2.m。

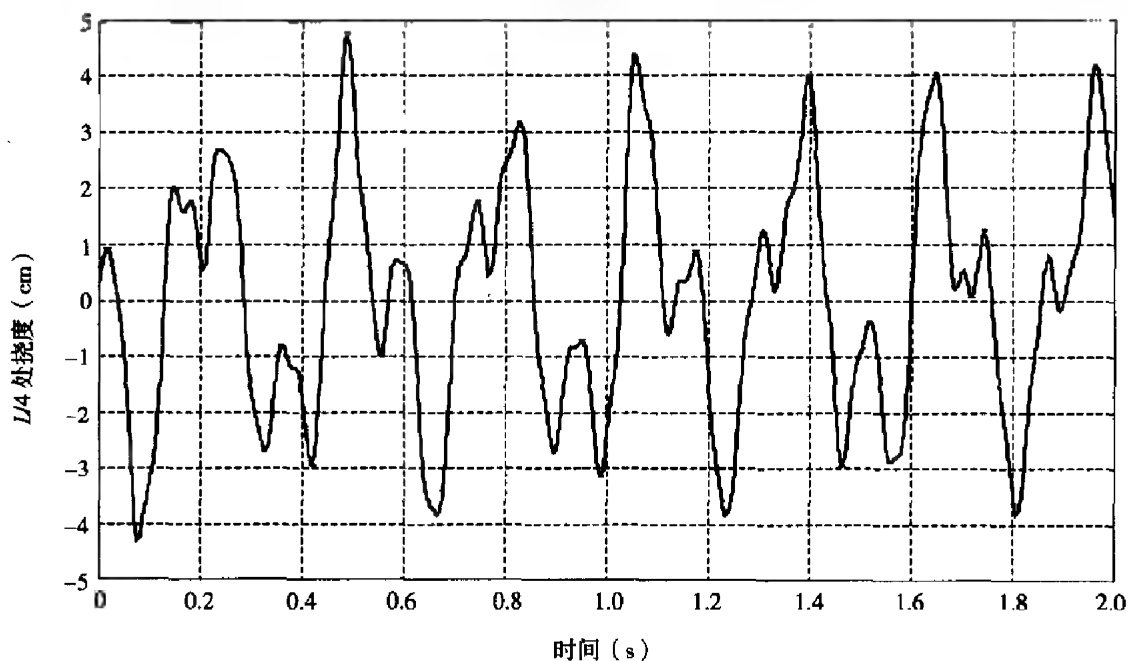


图 8-3 挠度（L/4 处）前两秒内时程曲线

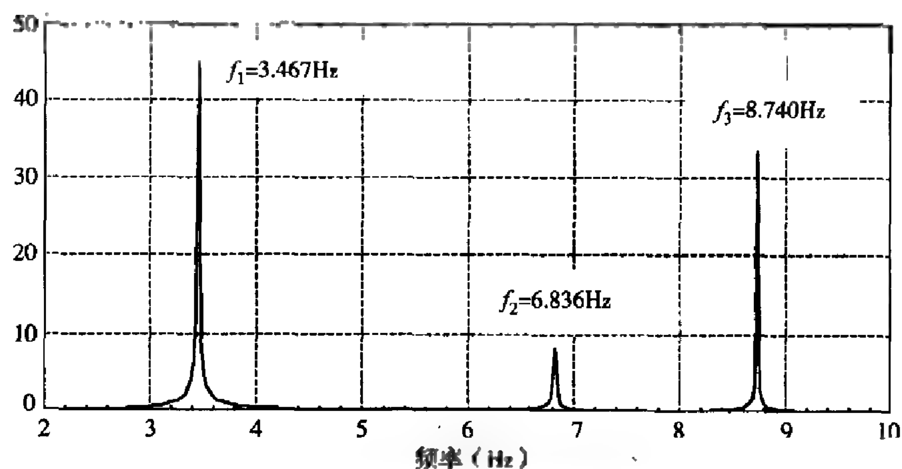


图 8-4 挠度时程曲线的频谱特性

参考文献

[1] Liu G. R. Quek S. S.（龙述尧，侯淑娟，钱长照，译）. 有限元法实用教程. 长沙：湖南大学出版社，2004.

[2] Zienkiewicz, O. C.. The Finite Element Method. 3rd ed. ,London:McGraw – Hill,1977.
[3] 朱伯芳. 有限单元法原理与应用. 2版. 北京：中国水利水电出版社,1998.
[4] 丁皓江,何福保,谢贻权,徐兴. 弹性与塑性力学中的有限单元法. 北京:机械工业出版社,1989.
[5] Bathe K. J. , Wilson E. L.. Numerical Method in Finite Element Method. New Jersey: Prentice Hall Inc. ,1976.
[6] Bathe K. J. , Wilson E. L.. Solution methods for eigenvalue problems in structural mechanics, International Journal for Numerical Methods in Engineering,1973,6(2): 213 –226.
[7] Wilkinson J. H.. The Algebraic Eigenvalue Problem. Oxford, UK:Clarendon Press,1965.
[8] Cheung Y. K. , Leung A. Y. T.. Finite Element Methods in Dynamics. Dordrecht:Kluwer Academic Publishers,1991.

21 世纪交通版

高等学校教材（公路与交通工程）编写委员会

总　　序

当今世界，科学技术突飞猛进，全球经济一体化趋势进一步加强，科技对于经济增长的作用日益显著，教育在国家经济与社会发展中所处的地位日益重要。进入新世纪，面对国际国内经济与社会发展所出现的新特点，我国的高等教育迎来了良好的发展机遇，同时也面临着巨大的挑战，高等教育的发展处在一个前所未有的重要时期：其一，加入WTO，中国经济已融入到世界经济发展的进程之中，国家间的竞争更趋激烈，竞争的焦点已更多地体现在高素质人才的竞争上，因此，高等教育所面临的是全球化条件下的综合竞争。其二，我国正处在由计划经济向社会主义市场经济过渡的重要历史时期，这一时期，我国经济结构调整将进一步深化，对外开放将进一步扩大，改革与实践必将提出许多过去不曾遇到的新问题，高等教育面临加速改革以适应国民经济进一步发展的需要。面对这样的形势与要求，党中央国务院提出扩大高等教育规模，着力提高高等教育的水平与质量。这是为中华民族自立于世界民族之林而采取的极其重大的战略步骤，同时，也是为国家未来的发展提供基础性的保证。

为适应高等教育改革与发展的需要，早在1998年7月，教育部就对高等学校本科专业目录进行了第四次全面修订。在新的专业目录中，土木工程专业扩大了涵盖面，原先的公路与城市道路工程、桥梁工程、隧道与地下工程等专业均纳入土木工程专业。本科专业目录的调整是为满足培养“宽口径”复合型人才的要求，对原有相关专业本科教学产生了积极的影响。这一调整是着眼于培养21世纪社会主义现代化建设人才的需要而进行的，面对新的变化，要求我们对人才的培养规格、培养模式、课程体系和内容都应作出适时调整，以适应要求。

根据形势的变化与高等教育所提出的新的要求，同时，也考虑到近些年来公路交通大发展所引发的需求，人民交通出版社通过对“八五”、“九五”期间的路桥及交通工程专业高校教材体系的分析，提出了组织编写一套面向21世纪的具有鲜明交通特色的高等学校教材的设想。这一设想，得到了原路桥教学指导委员会几乎所有成员学校的广泛响应与支持。2000年6月，由人民交通出版社发起组织全国面向交通办学的12所高校的专家学者组成面向21世纪交通版高等学校教材（公

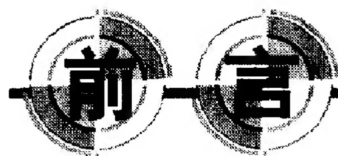
路类）编审委员会，并召开第一次会议，会议决定着手组织编写土木工程专业具有交通特色的道路专业方向、桥梁专业方向以及交通工程专业教材。会议经过充分研讨，确定了包括基本知识技能培养层次、知识技能拓宽与提高层次以及教学辅助层次在内的约 130 种教材，范围涵盖本科与研究生用教材。会后，人民交通出版社开始了细致的教材编写组织工作，经过自由申报及专家推荐的方式，近 20 所高校的百余名教授承担约 130 种教材的主编工作。2001 年 6 月，教材编委会召开第二次会议，全面审定了各门教材主编院校提交的教学大纲，之后，编写工作全面展开。

面向 21 世纪交通版高等学校教材编写工作是在本科专业目录调整及交通大发展的背景下展开的。教材编写的基本思路是：(1)顺应高等教育改革的形势，专业基础课教学内容实现与土木工程专业打通，同时保留原专业的主干课程，既顺应向土木工程专业过渡的需要，又保持服务公路交通的特色，适应宽口径复合型人才培养的需要。(2)注重学生基本素质、基本能力的培养，将教材区分为二个主层次与一个辅助层次，即基本知识技能培养层次与知识技能拓宽与提高层次，辅助层次为教学参考用书。工作的着力点放在基本知识技能培养层次教材的编写上。(3)目前，中国的经济发展存在地区间的不平衡，各高校之间的发展也不平衡，因此，教材的编写要充分考虑各校人才培养规格及教学需求多样性的要求，尽可能为各校教学的开展提供一个多层次、系统而全面的教材供给平台。(4)教材的编写在总结“八五”“九五”工作经验的基础上，注意体现原创性内容，把握好技术发展与教学需要的关系，努力体现教育面向现代化、面向世界、面向未来的要求，着力提高学生的创新思维能力，使所编教材达到先进性与实用性兼备。(5)配合现代化教学手段的发展，积极配套相应的教学辅件，便利教学。

教材建设是教学改革的重要环节之一，全面做好教材建设工作，是提高教学质量的重要保证。本套教材是由人民交通出版社组织，由原全国高等学校路桥与交通工程教学指导委员会成员学校相互协作编写的一套具有交通出版社品牌的教材，教材力求反映交通科技发展的先进水平，力求符合高等教育的基本规律。各门教材的主编均通过自由申报与专家推荐相结合的方式确定，他们都是各校相关学科的骨干，在长期的教学与科研实践中积累了丰富的经验。由他们担纲主编，能够充分体现教材的先进性与实用性。本套教材预计在二年内完全出齐，随后，将根据情况的变化而适时更新。相信这批教材的出版，对于土木工程框架下道路工程、桥梁工程专业方向与交通工程专业教材的建设将起到有力的促进作用，同时，也使各校在教材选用方面具有更大的空间。需要指出的是，该批教材中研究生教材占有较大比例，研究生教材多具有较高的理论水平，因此，该套教材不仅对在校学生，同时对于在职学习人员及工程技术人员也具有很好的参考价值。

21 世纪初叶，是我国社会经济发展的重要时期，同时也是我国公路交通从紧张和制约状况实现全面改善的关键时期，公路基础设施的建设仍是今后一项重要而艰巨的任务，希望通过各相关院校及所有参编人员的共同努力，尽快使全套面向21 世纪交通版高等学校教材（公路类）尽早面世，为我国交通事业的发展做出贡献。

21 世纪交通版
高等学校教材(公路类)编审委员会
人民交通出版社
2001 年 12 月



前言 PREFACE

有限元法经过半个世纪的发展，在理论和实践上均取得了引人瞩目的成就，事实上它已经发展成为工程领域中一门不可或缺的技术，同时也是科技工作者进行科学研究的有力工具。因此，有限元法逐渐成为高等院校理工科专业的必修课程。

本书是编者在为浙江大学桥梁与隧道专业硕士生所讲授的《结构分析的有限元法》课程讲义的基础上编写而成的。本书的一大特色是采用 **MATLAB** 作为编程平台，利用 **MATLAB** 强大的科学计算和符号运算功能，帮助读者轻松跨越繁琐的公式推导和复杂的编程技巧，获得最佳的学习效率。

在写作上，采用理论和程序紧密结合的方法，以增加读者的感性认识，更好地理解有限元理论。不仅每章后面有极强工程背景的数值算例和源程序，而且还在有限元理论的叙述过程中，插入相应的 **MATLAB** 程序段以帮助理解。

本书的内容共分 **8** 章，第 **1**、第 **2** 章主要讲述有限元和 **MATLAB** 的发展概况及其基础知识；第 **3** 章介绍了在工程中有极大应用价值的杆系结构的各类单元，包括杆单元、平面和空间梁单元；第 **4** 章讲述经典的平面问题三角形单元和四边形单元；第 **5** 章把第 **4** 章的平面单元推广到空间单元，还讨论了空间轴对称单元；第 **6** 章介绍了应用最广的平面和空间等参数单元以及轴对称等参数单元；第 **7** 章在简要介绍板理论的基础上，讨论了三角形和矩形薄板单元，也给出了基于 **Mindlin** 板理论的四边形板单元；最后介绍了用平板单元和平面应力单元组合而成的壳单元；第 **8** 章讲述了结构的动力和稳定问题，并介绍了几种常用的特征值问题和动力响应问题的算法。

本书在写作过程中，得到了浙江大学丁皓江教授和徐兴教授的指导，他们详细审阅了初稿，指出并修正了若干错误，在此对他们表示衷心的感谢。本书还得到了浙江大学项贻强教授、陈伟球教授、叶贵如教授、蔡金标副教授、赵阳讲师和张治成博士的指导和帮助。我要特别感谢我的家人，感谢他们在我写作此书时所做的牺牲。

由于编者水平有限，时间仓促，书中难免有许多缺点甚至错误，热情欢迎专家和读者批评指正，任何意见和建议均可通过电子邮箱 xurongqiao@zju.edu.cn 发给编者。读者也可通过这个邮箱索取本书的源程序和数据文件。

徐荣桥

2005 年 12 月

21世纪交通版研究生教学用书

道路工程方向

高等土质学	长安大学
路线设计原理	华南理工大学　长安大学
路基设计原理	长安大学　长沙理工大学
路面设计原理	同济大学　长沙理工大学
现代道路工程材料	华南理工大学　重庆交通大学
路线 CAD 原理与方法	东南大学
道路规划与几何设计	同济大学
道路工程结构分析数值方法	长沙理工大学
路面断裂与损伤	长安大学
沥青与沥青混合料	长安大学
路面功能设计	东南大学
道路景观设计	东南大学
路面新技术	东南大学

桥梁与隧道工程方向

高等桥梁结构理论	同济大学
高等隧道与地下工程结构理论	同济大学　长安大学　重庆交通大学
高等桥梁钢结构基本原理	同济大学
高等混凝土结构理论	湖南大学　哈尔滨工业大学
高等工程结构试验	同济大学
结构工程数值方法	长安大学
结构分析的有限元法与 MATLAB 程序设计	浙江大学
现代钢桥(上)	同济大学
桥梁结构电算	同济大学
箱梁理论（第二版）	福州大学
桥梁施工控制与监测	长沙理工大学　湖南大学　重庆交通大学
现代预应力技术理论与实践	福州大学　重庆交通大学
轨道交通桥梁结构理论	同济大学　重庆交通大学
桥梁健康状态监测	同济大学